

KLASIFIKASI DAN REKOMENDASI JURUSAN KULIAH BAGI PELAJAR SMA MENGGUNAKAN ALGORITME *NAÏVE BAYES*- WP

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Restu Fitriawanti
NIM: 145150200111126



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

KLASIFIKASI DAN REKOMENDASI JURUSAN KULIAH BAGI PELAJAR SMA
MENGUNAKAN ALGORITME NAÏVE BAYES-WP

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Restu Fitriawanti
NIM: 145150200111126

Skripsi ini telah diuji dan dinyatakan lulus pada
06 Juni 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Imam Cholissodin, S.Si, M.Kom
NIK. 201201 850719 1 001


Ratih Kartika Dewi, S.T., M.Kom
NIK. 201503 890520 2 001

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 6 Mei 2018



Restu Fitriawanti

NIM: 145150200111126

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena berkat Rahmat dan Karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“Klasifikasi Dan Rekomendasi Jurusan Kuliah Bagi Pelajar SMA Menggunakan Algoritme Naïve Bayes-WP”**. Shalawat beserta salam semoga senantiasa terlimpah curahkan kepada Nabi Muhammad SAW, kepada keluarganya, para sahabatnya, hingga kepada umatnya hingga akhir zaman, amin. Penulisan tugas akhir ini diajukan untuk memenuhi salah satu syarat untuk menyelesaikan studi di Jurusan Informatika, Fakultas Ilmu Komputer Universitas Brawijaya.

Dalam penyusunan dan penulisan seminar skripsi ini tidak terlepas dari bantuan, bimbingan serta dukkungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis dengan senang hati menyampaikan terima kasih kepada yang terhormat :

1. Imam Cholissodin, S.Si, M.Kom selaku pembimbing I dan Ratih Kartika Dewi, S.T., M.Kom selaku pembimbing II yang telah memberikan arahan petunjuk dan bimbingannya dalam penyelesaian skripsi ini.
2. Bapak Agus Wahyu Widodo, S.T,M.Sc selaku Ketua Program Studi Jurusan Informatika.
3. Bapak Tri Astoto, S.T,M.T,Ph.D selaku Ketua Jurusan Teknik Informatika
4. Orang tua penulis, Bapak Bambang Iwan S dan Ibu Sulastri yang tak henti-hentinya memberikan dukungan moril dan materil.
5. Saudara penulis, Riesta Astriawanti dan Reski Tri Setyawan yang telah memberikan semangat dan motivasi untuk terselesaikannya skripsi ini.
6. Seluruh dosen Fakultas Ilmu Komputer Universitas Brawijaya atas kesediaan membagi dan memberi ilmu kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
7. Alvin Erzal Syahreza yang telah memberikan dukungan mental dan semangat kepada saya untuk menyelesaikan penulisan skripsi ini.
8. Sahabat Penulis sejak awal kuliah hingga akhir kuliah yaitu sevenstar (Eno, Arig, Kemas, Reza, Alif dan Rey)dan cumlaude (Nanda, Ficry, Aji, Agri, Angky, Yogi, Dimas dan Adit) yang telah memberi saya semangat dalam menyelesaikan skripsi ini.
9. Teman-teman seperjuangan Teknik Informatika angkatan 2014 selaku teman satu angkatan yangtelah mendukung dan berjuang bersama untuk mencapai gelar sarjana.

10. Dan semua pihak yang tidak bisa disebutkan satu per satu. Terima kasih atas segala bantuannya.

Semoga Allah SWT memberikan balasan yang berlipat ganda kepada semuanya. Demi perbaikan selanjutnya, saran dan kritik yang membangun akan penulis terima dengan senang hati.

Malang, 5 Mei 2018

Restu Fitriawanti



ABSTRAK

Tiap tahunnya pelajar SMA semester akhir akan dihadapkan oleh sebuah pilihan untuk menentukan jurusan apa yang akan di pilih untuk pendidikannya kelak. Setiap pilihan akan menentukan masa depan pemilihnya, dan ini adalah sesuatu yang cukup sulit untuk di tentukan oleh kebanyakan pelajar SMA, karena tidak memiliki informasi dan gambaran terkait dengan pendidikan di perguruan tinggi. Selain itu, anak tersebut masih belum mengetahui minat dan kemampuan pada dirinya. Berdasarkan permasalahan diatas pemilihan jurusan sedini mungkin harus mulai dipertimbangkan karena memilih fakultas dan jurusan dengan tepat sangatlah sulit, jika salah memilih jurusan akan mengakibatkan pelajar terhambat dalam proses pembelajaran di perkuliahannya, karena kurang nyaman dengan materi-materi di perkuliahannya dan mungkin banyak materi-materi yang kurang di sukai. Hal ini akan berpengaruh pada indeks prestasi (IP) anak tersebut yang bisa berada di bawah standar dan yang lebih buruk lagi yaitu dikeluarkannya pelajar (DO-Drop Out) karena dinyatakan tidak mampu mengikuti pendidikan yang diikutinya. Maka klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA yang berdasarkan nilai akademik aja dapat membantu pelajar SMA untuk menentukan pilihanya dengan tepat. Perhitungan penelitian dihitung secara terpisah untuk algoritme Naïve Bayes digunakan untuk mengklasifikasi data nilai pelajar ke dalam kelas fakultas dan Weighted Product (WP) digunakan untuk membantu menentukan jurusan yang tepat berdasarkan jurusan di fakultas yang telah ditentukan sebelumnya oleh algoritme Naïve Bayes. Dengan menggunakan algoritme Naïve Bayes-WP maka dihasilkan rata-rata akurasi sistem mencapai 82%.

Kata Kunci : *Naïve Bayes*, *Weighted Product*, Rekomendasi Jurusan, Klasifikasi

ABSTRACT

Each year high school students will be faced with a final choice to determine what direction will be selected for future education. Each choice will determine the future of the voter, and this is something that is difficult enough to be determined by most high school students, because they do not have information and images related to education in college. In addition, the child is still not aware of the interests and abilities on him. Based on the problems above the selection of majors as early as possible should start to be considered because choosing faculty and majors with precisely very difficult, if one chose the department will result in learner in the learning process in lectures, because less comfortable with the materials in the lecture and probably a lot of less-liked material. This will affect the child's achievement index (IP) that can be below the standard and worse the discharge of the student (DO-Drop Out) because it is declared not able to follow the education that followed. So the classification and recommendation of college majors for high school students who based on academic grades wrote can help high school students to determine the proper choice. The calculation of the study is calculated separately for the Naïve Bayes algorithm used to classify student learner data into the faculty class and Weighted Product (WP) is used to help determine the exact majors based on the majors in the faculty predetermined by the Naïve Bayes algorithm. By using the Naïve Bayes-WP algorithm, the system's average accuracy reaches 82%.

Keywords: Naïve Bayes, Weighted Product, Recommendation of Department, Classification

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR <i>SOURCE CODE</i>	xiii
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka.....	6
2.2 Dasar Teori.....	10
2.2.1 Jurusan Kuliah	10
2.2.2 Klasifikasi.....	12
2.2.3 <i>Naïve Bayes</i>	12
2.2.4 <i>Weighted Product</i> (WP)	14
2.2.5 Akurasi.....	15
BAB 3 METODOLOGI	16
3.1 Tahap Penelitian	16
3.2 Studi Pustaka	16

3.3 Pengumpulan Data	17
3.4 Analisis Kebutuhan	17
3.5 Perancangan Sistem	17
3.6 Pengujian Algoritme	18
3.6.1 Pengujian Jumlah Data Latih dan Data Uji Pada Algoritme <i>Naïve Bayes</i>	18
3.6.2 Pengujian Jumlah Data Latih dan Data Uji Pada Algoritme <i>Naïve Bayes</i> dengan Data Latih Diskrit	19
3.6.3 Pengujian Bobot Nilai Minimum Tiap Jurusan pada Algoritme <i>Weighted Product</i> (WP)	19
3.7 Kesimpulan	20
BAB 4 PERANCANGAN.....	21
4.1 Penyelesaian Masalah Menggunakan <i>Naïve Bayes</i> -WP	21
4.1.1 Penyelesaian dengan <i>Naïve Bayes</i>	22
4.1.2 Penyelesaian dengan <i>Weighted Product</i> (WP)	29
4.2 Manualisasi Menggunakan Metode <i>Naïve Bayes</i> -WP.....	37
4.2.1 Manualisasi <i>Naïve Bayes</i>	37
4.2.2 Manualisasi <i>Weighted Product</i> (WP)	41
4.3 Perancangan Antar Muka	44
4.3.1 Perancangan Tampilan Utama.....	45
4.3.2 Perancangan Tampilan Daftar Data	45
4.3.3 Perancangan Tampilan Pengaturan Data Uji	46
4.3.4 Perancangan Tampilan Pengujian <i>Naïve Bayes</i>	46
4.3.5 Perancangan Tampilan Manualisasi <i>Naïve Bayes</i>	47
4.3.6 Perancangan Tampilan Pengujian <i>Weighted Product</i> (WP)	47
4.3.7 Perancangan Tampilan Manualisasi <i>Weighted Product</i> (WP)	48
4.3.8 Uji Coba Jumlah Data Latih dan Data Uji Algoritme <i>Naïve Bayes</i>	48
4.3.9 Uji Coba Jumlah Data Latih dan Data Uji Algoritme <i>Naïve Bayes</i> dengan Data Diskrit.....	49
4.3.10 Uji Coba Bobot Nilai Minimum Tiap Jurusan pada Algoritme <i>Weighted Product</i> (WP)	49

BAB 5 IMPLEMENTASI	51
5.1 Spesifikasi Sistem	51
5.1.1 Spesifikasi Perangkat Keras	51
5.1.2 Spesifikasi Perangkat Lunak	51
5.1.3 Batasan-Batasan Implementasi	51
5.2 Implementasi Algoritme	52
5.2.1 Implementasi Algoritme <i>Naïve Bayes</i>	52
5.2.2 Implementasi Algoritme <i>Weighted Product</i> (WP)	58
5.3 Implementasi Antar Muka	68
5.3.1 Implementasi Tampilan Utama	68
5.3.2 Implementasi Tampilan Daftar Data	69
5.3.3 Implementasi Tampilan Data Uji	69
5.3.4 Implementasi Tampilan Pengujian <i>Naïve Bayes</i>	70
5.3.5 Implementasi Tampilan Manualisasi <i>Naïve Bayes</i>	70
5.3.6 Implementasi Tampilan Pengujian <i>Weighted Product</i> (WP)	71
5.3.7 Implementasi Tampilan Manualisasi <i>Weighted Product</i> (WP)	71
BAB 6 PENGUJIAN DAN ANALISIS	72
6.1 Pengujian dan Analisis Jumlah Data Latih dan Data Uji Algoritme <i>Naïve Bayes</i>	72
6.2 Pengujian dan Analisi Jumlah Data Latih dan Data Uji Algoritme <i>Naïve Bayes</i> dengan Data Diskrit	73
6.3 Pengujian dan Analaisis Pengujian Bobot Nilai Minimum Tiap Jurusan Pada Algoritme <i>Weighted Product</i> (WP)	75
BAB 7 PENUTUP	79
7.1 Kesimpulan	79
7.2 Saran	79
daftar pustaka	81
LAMPIRAN 1 SURAT PERMOHONAN DATA SKRIPSI	83
LAMPIRAN 2 SURAT REKOMENDASI DINAS PENDIDIKAN	84
LAMPIRAN 3 DATA NILAI PELAJAR SMA	85
LAMPIRAN 4 PEMETAAN KEMINATAN JURUSAN KULIAH	135

DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Klasifikasi Untuk Menentukan Pilihan Jurusan Kuliah Bagi Pelajar SMA dengan Menggunakan Algoritme Naïve Bayes-WP.....	16
Gambar 4.1 Diagram Alir Algoritme Penyelesaian Masalah.....	21
Gambar 4.2 Diagram Alir Penyelesaian Naïve Bayes.....	22
Gambar 4.3 Diagram Alir Pelatihan	23
Gambar 4.4 Diagram Alir Pengujian.....	24
Gambar 4.5 Daftar Kelas Hasil Klasifikasi Algoritme Naïve Bayes	24
Gambar 4.6 Diagram Perhitungan Prior	25
Gambar 4.7 Diagram Alir Perhitungan Likelihood	26
Gambar 4.8 Menghitung Mean	27
Gambar 4.9 Menghitung Standar Deviasi	28
Gambar 4.10 Diagram Alir Perhitungan Posterior	29
Gambar 4.11 Diagram Alir Algoritme Penyelesaian Masalah.....	30
Gambar 4.12 Diagram Alir Penentuan Bobot tiap Mata Pelajaran	32
Gambar 4.13 Diagram Alir Penentuan Bobot Tiap Jurusan	34
Gambar 4.14 Diagram Alir Perbaikan Bobot (W_i)	35
Gambar 4.15 Diagram Alir Menghitung Vektor S	36
Gambar 4.16 Diagram Alir Menghitung Vektor V	37
Gambar 4.17 PerancanganTampilan Utama.....	45
Gambar 4.18 PerancanganTampilan DaftarData.....	45
Gambar 4.19 PerancanganTampilan Pengaturan Data Uji.....	46
Gambar 4.20 PerancanganTampilan Pengujian <i>Naïve Bayes</i>	46
Gambar 4.21 Perancangan Tampilan Manualisasi <i>Naïve Bayes</i>	47
Gambar 4.22 PerancanganTampilan Pengujian WP	47
Gambar 4.23 PerancanganTampilan Manualisasi WP.....	48
Gambar 5.1 Implementasi Tampilan Utama.....	68
Gambar 5.2 Implementasi Tampilan Daftar Data	69
Gambar 5.3 Implementasi Tampilan Data Uji.....	69

Gambar 5.4 Implementasi Tampilan Pengujian <i>Naïve Bayes</i>	70
Gambar 5.5 Implementasi Tampilan Manualisasi <i>Naïve Bayes</i>	70
Gambar 5.6 Implementasi Tampilan Pengujian WP	71
Gambar 5.7 Tampilan Manualisasi WP	71
Gambar 6.1 Grafik Hasil Rata-Rata Akurasi Jumlah Data Latih dan Data Uji.....	73
Gambar 6.2 Grafik Hasil Rata-Rata Akurasi Uji Coba Jumlah Data dengan Data Latih dan Data Uji Bertipe Diskrit	75
Gambar 6.3 Grafik Hasil Rata-Rata Akurasi Uji Coba bobot nilai minimum tiap jurusan pada algoritme WP	76
Gambar 6.4 Grafik Hasil Nilai Akurasi Tertinggi Tiap Variasi Pada Uji Coba bobot nilai minimum tiap jurusan pada algoritme WP	78



DAFTAR SOURCE CODE

<i>Source Code</i> 5.1 Pendefinisian Data	53
<i>Source Code</i> 5.2 Perhitungan <i>Naïve Bayes</i>	56
<i>Source Code</i> 5.3 Pengambilan Data Nilai Calon Mahasiswa	59
<i>Source Code</i> 5.4 Pengambilan Data Nilai Minimum	61
<i>Source Code</i> 5.5 Pembobotan Mata Pelajaran untuk Tiap Fakultas.....	63
<i>Source Code</i> 5.6 Perhitungan WP	65
<i>Source Code</i> 5.7 Penentuan Hasil Rekomendasi.....	67



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	7
Tabel 2.2 Nilai Pembobotan Jurusan	15
Tabel 4.1 Variasi bobot nilai minimum tiap jurusan	30
Tabel 4.2 Aturan Pembobotan	31
Tabel 4.3 Keterangan Bobot	31
Tabel 4.4 Data Uji Sampel	38
Tabel 4.5 Data Uji Sampel	39
Tabel 4.6 Likelihood Atribut untuk Peluang Diterima	39
Tabel 4.7 Likelihood Atribut untuk Peluang Ditolak	40
Tabel 4.8 Data Uji Sampel	41
Tabel 4.9 Bobot Data Uji Sampel	42
Tabel 4.10 Nilai Minimum tiap Jurusan	42
Tabel 4.11 Bobot Perbandingan Nilai Calon Mahasiswa dengan Nilai Minimum Jurusan	42
Tabel 4.12 Perbaikan Bobot Kriteria(W_i)	43
Tabel 4.13 Menghitung Vektor S	44
Tabel 4.14 Menghitung Vektor V	44
Tabel 4.15 Data Uji Sampel	48
Tabel 4.16 Data Uji Sampel dengan Mengubah menjadi Data Diskrit	49
Tabel 4.17 Data Uji Perubahan Bobot	50
Tabel 6.1 Hasil Uji Coba Jumlah Data Latih dan Data Uji Algoritme <i>Naïve Bayes</i>	72
Tabel 6.2 Keterangan <i>Range</i> Nilai	73
Tabel 6.3 Hasil Pengolahan Data Kontinu ke Data Diskrit	74
Tabel 6.4 Hasil Uji Coba Jumlah Data Latih dan Data Uji Algoritme <i>Naïve Bayes</i> dengan Data Diskrit	74
Tabel 6.5 Hasil Uji Coba Pengujian bobot nilai minimum tiap jurusan pada algoritme WP	76
Tabel 6.6 Hasil Nilai Tertinggi Pada Pengujian bobot nilai minimum tiap jurusan pada algoritme WP	77

DAFTAR LAMPIRAN

LAMPIRAN 1 SURAT PERMOHONAN DATA SKRIPSI	83
LAMPIRAN 2 SURAT REKOMENDASI DINAS PENDIDIKAN	84
LAMPIRAN 3 DATA NILAI PELAJAR SMA.....	85
LAMPIRAN 4 PEMETAAN KEMINATAN JURUSAN KULIAH	135



BAB 1 PENDAHULUAN

1.1 Latar belakang

Bukan perkara mudah bagi para siswa Sekolah Menengah Atas (SMA) yang akan lulus dalam menentukan perguruan tinggi mana dan jurusan apa yang harus dipilih. Apa pun keputusan yang diambil, hal itu menjadi pinjakan awal yang akan menentukan nasib dan masa depan mereka. Namun, banyak siswa yang berada di tahun terakhir SMA masih belum mengetahui minat dan kemampuannya sehingga masih gagap menerka jurusan kuliahnya (Prasetya, et al., 2016).

Jika melihat data statistik nasional yang dikelola Kementerian Pendidikan Nasional, tampak peningkatan jumlah pendaftaran di perguruan tinggi, baik negeri maupun swasta. Di tahun 2017 jumlah peserta yang mengikuti Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) 2017 mencapai 797.738 pendaftar. Jumlah ini mengisyaratkan kenaikan jumlah peserta SBMPTN sebanyak 76.412 pendaftar dibanding tahun lalu. Peningkatan juga terjadi pada bagi peserta Bidikmisi yang mengikuti SBMPTN 2017, tahun lalu sebanyak 124.398 peserta sedangkan tahun ini ada 158.157 peserta. Sedangkan, jumlah pendaftar Seleksi Nasional Penerimaan Mahasiswa Perguruan Tinggi Negeri (SNMPTN) 2017 berjumlah 517.418 siswa. Data ini menunjukkan penurunan jumlah pendaftar dibandingkan dengan tahun-tahun sebelumnya. Pada 2016 jumlah pendaftar SNMPTN mencapai 645.202 siswa. Sementara pada 2015 peserta mencapai 852.093. Walaupun data menunjukkan jumlah penurunan namun bukan berarti peminat peserta untuk mengikuti seleksi jalur SNMPTN menurun, hal ini disebabkan diterapkan aturan baru persyaratan pendaftaran SNMPTN bagi SMA sederajat yang didasari pada akreditasi. Berdasarkan fakta tersebut, menunjukkan bahwa untuk melanjutkan pendidikan ke perguruan tinggi bukan hal mudah dimana jumlah persaingan setiap tahun meningkat. Sehingga untuk melanjutkan pendidikan ke perguruan tinggi harus menerapkan strategi yang tepat (Ramdhani, 2017).

Strategi awal yang harus diterapkan adalah mengetahui kemampuan dan bakat masing-masing. Dengan memiliki kemampuan dan bakat yang sesuai maka seorang pelajar akan lebih menikmati proses seleksi yang akan dijalani sehingga setiap proses dapat dilalui dengan baik yang diharapkan mendapatkan hasil yang lebih memuaskan. Sehingga setelah diterima di perguruan tinggi pun para pelajar dapat menikmati masa kuliah dengan jurusan yang benar-benar disukai, karena jika tidak kuliah tersebut akan berjalan sebagai beban (Kumala, et al., 2015).

Berdasarkan penelitian Pemilihan jurusan kuliah yang kurang tepat biasanya diakibatkan beberapa alasan seperti gengsi, ikut-ikutan teman, saran orang tua, atau tidak ada pilihan jurusan, hal tersebut akan mengakibatkan pelajar

terhambat dalam proses pembelajaran di perkuliaannya, karena kurang nyaman dengan materi-materi di perkuliaannya dan mungkin banyak materi-materi yang kurang di sukai. Hal ini akan berpengaruh pada indeks prestasi (IP) anak tersebut yang bisa berada di bawah standar dan yang lebih buruk lagi yaitu dikeluarkannya pelajar (DO-Drop Out) karena dinyatakan tidak mampu mengikuti pendidikan yang diikutinya (Kumala, 2015).

Sehingga pemilihan jurusan harus dilakukan sedini mungkin dengan penuh pertimbangan. Salah memilih jurusan dapat menyebabkan kerugian yang besar bagi calon Mahasiswa baru di masa depan. Oleh karena itu, calon mahasiswa baru harus mampu mengetahui kemampuan dan minat yang dimilikinya. Salah satu cara yang dapat membantu calon Mahasiswa baru untuk mengetahui jurusan yang sesuai dengan kemampuan dan potensinya adalah ketersediaan sistem klasifikasi penentuan jurusan kuliah untuk pelajar SMA berdasarkan nilai akademis.

Dalam menentukan suatu pilihan, dapat dilakukan secara matematis dengan beberapa metode. Penelitian yang dilakukan oleh Rahmadi Wijaya (2014) tentang membuat perancangan sistem untuk pemilihan lokasi perumahan menggunakan metode *Weighted Product* (WP). Sistem ini dibuat untuk mempermudah dalam pemilihan lokasi perumahan, dengan menggunakan metode *Weighted Product* (WP) dapat membantu menyelesaikan permasalahan tersebut karena pemilihan rumah dapat di golongankan ke dalam masalah yang bersifat *multiobjectives* (ada banyak tujuan yang ingin dicapai) dan *multicriteria* (ada banyak kriteria untuk mencapai tujuan). Penelitian lain yang dilakukan oleh Mega Esti (2016) mengenai pembuatan sistem penerimaan BPJS menggunakan metode *Naïve Bayes*. Penelitian ini dibuat untuk menentukan penerima BPJS dengan menggunakan kriteria dengan metode *Naïve Bayes*. Metode *Naïve Bayes Classifier* (NBC) merupakan teknik prediksi berbasis *probabilistic* sederhana yang berdasar pada penerapan teorema bayes (atau aturan bayes) dengan asumsi independensi (ketidak ketergantungan) yang kuat (naïf) (Prasetyo, 2012). Metode *Naïve Bayes* memiliki akurasi yang sangat tinggi, sehingga keputusan yang direkomendasikan pada rancangan sistem ini akan lebih cepat, tepat dan akurat (Prasetyo, 2012). Berdasarkan uraian diatas maka penelitian ini dilakukan yaitu membuat dan membahas sistem klasifikasi untuk membantupelajar SMA memilih jurusan kuliah yang tepat berdasarkan beberapa factor dan variabel dengan menggunakan metode Algoritme *Naïve Bayes-WP*, karena metode tersebut memiliki keunggulan dalam pembobotan kriteria yang secara otomatis tanpa campur ahli tangan, sehingga proses penentuan hasil berjalan secara cepat dan dinamis. Penelitian ini juga mencoba untuk mengklasifikasi beberapa pelajar kedalam kategori jurusan yang cocok.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dipaparkan secara jelas diatas, maka rumusan masalah ditekankan pada :

1. Bagaimana menerapkan algoritme *Naïve Bayes*-WP ke dalam klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA.
2. Bagaimana tingkat akurasi klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA.

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dipaparkan, adapun penelitian ini bertujuan untuk.

1. Menerapkan algoritme *Naïve Bayes* dan algoritme *Weighted Product* (WP) ke dalam klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA.
2. Menguji tingkat akurasi klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA dengan menggunakan Algoritme *Naïve Bayes*-WP.

1.4 Manfaat

Adapun manfaat yang diperoleh dari penelitian ini diantara lain:

1. Untuk Pelajar SMA :
 - Diharapkan dapat mempermudah pelajar SMA dalam pemilihan jurusan kuliah yang tepat bagi dirinya.
2. Untuk Penulis :
 - Dapat menerapkan ilmu tentang algoritme *Naïve Bayes* dan algoritme *Weighted Product* (WP)
 - Dapat mengetahui tingkat akurasi terbaik pada Algoritme *Naïve Bayes* dan Algoritme *Weighted Product* (WP)
 - Dapat mengembangkan sistem klasifikasi terbaru ke depannya dengan metode yang sudah ada maupun dengan metode baru.
3. Untuk Sekolah Menengah Atas :
 - Dapat membantu guru bimbingan konseling (BK) di SMA untuk mengarahkan siswanya kepada satu jurusan kuliah yang tepat.
4. Untuk Universitas :
 - Dapat membantu Universitas mendapatkan calon Mahasiswa yang tepat untuk setiap jurusan sehingga kedepannya Mahasiswa tersebut dapat memberikan kemajuan untuk Universitasnya.

1.5 Batasan masalah

Beberapa hal yang menjadi batasan masalah dalam pembuatan sistem ini antara lain :

1. Data latih dan data uji yang digunakan dalam penelitian ini berasal dari data mata pelajaran yang berasal dari siswa SMAN 3 Malang pada 2 tahun terakhir.
2. Penentuan rekomendasi jurusan kuliah hanya diproses berdasarkan nilai rapor siswa semester 3 hingga semester 5.
3. Data yang digunakan hanya berdasarkan dari data mata pelajaran tanpa mempertimbangkan universitas dan kuota jurusan tersebut.
4. Proses perhitungan menggunakan algoritme *Naïve Bayes* dan algoritme *Weighted Product* (WP) dilakukan secara terpisah. *Naïve Bayes* digunakan untuk menentukan fakultas dan *Weighted Product* (WP) digunakan untuk menentukan jurusan.

1.6 Sistematika pembahasan

Sistematika pembahasan penelitian ini disusun untuk memberikan gambaran umum dan uraian dari penyusunan tugas akhir secara garis besar. Sistematika pembahasan ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat dan batasan masalah pada klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme *Naïve Bayes*-WP.

BAB II LANDASAN KEPUSTAKAAN

Bab ini membahas mengenai kajian pustaka dan dasar teori yang berkaitan. Kajian pustaka membahas pemaparan singkat penelitian yang sudah pernah dilakukan namun terkait dengan penelitian klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme *Naïve Bayes*-WP. Dasar teori membahas tentang teori-teori yang mendukung dalam pengembangan dan perancangan penelitian klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme *Naïve Bayes*-WP.

BAB III METODOLOGI

Bab ini membahas metode-metode yang digunakan perancangan dan langkah kerja yang digunakan dalam membangun penelitian klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme *Naïve Bayes*-WP. Metodologi penelitian

yang digunakan terdiri dari studi pustaka, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisis dan kesimpulan.

BAB IV PERANCANGAN

Bab ini membahas perancangan sesuai kebutuhan dari sistem klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme Naïve Bayes-WP, kemudian merancang sistem sesuai dengan kebutuhan.

BAB V IMPLEMENTASI

Bab ini membahas implementasi dan pembahasan klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme Naïve Bayes-WP.

BAB VI PENGUJIAN DAN ANALISIS

Bab ini membahas tentang rencana pengujian yang akan dilakukan pada klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme Naïve Bayes-WP yang telah di implementasikan.

BAB VII PENUTUP

Bab ini merupakan salah satu bab yang membahas kesimpulan dari hasil penelitian dan saran untuk pengembangan sistem selanjutnya agar lebih baik.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisikan tentang landasan pustaka dan beberapa dasar teori yang berhubungan dengan penelitian skripsi ini secara umum. Landasan pustaka membahas teori-teori yang mengenai penjurusan keminatan, klasifikasi, algoritme *Naïve Bayes*, algoritme *Weighted Product* (WP).

2.1 Kajian Pustaka

Penelitian mengenai pengklasifikasi bukanlah baru pertama kali dilakukan, sudah ada beberapa penelitian terdahulu yang menerapkannya menggunakan Algoritme *Naïve Bayes* dan *Weighted Product*. Pada penelitian metode *Naïve Bayes* digunakan untuk merekomendasikan calon sekolah terbaik bagi siswa dengan menggunakan data sekolah menengah atas di Kota Sragen. Untuk mencari probabilitas terbesar dari semua semua instance pada atribut target seperti jarak, biaya SPP, akreditasi, tingkat kelulusan dan nilai rata-rata minimal UAN. Hasil dari penelitian ini dapat membantu menyajikan informasi sekolah menengah keatas yang sesuai dengan kriteria. Sehingga dapat menjadi pertimbangan bagi siswa untuk memilih calon sekolah menengah atas yang terbaik sesuai dengan kriteria yang diajukan (Salma, 2016).

Penelitian klasifikasi selanjutnya dengan menggunakan metode *Naïve Bayes* pada simulasi *Hadoop Multinode Cluster* untuk seleksi asisten praktikum dengan cara melakukan hasil klasifikasi tertinggi terhadap hasil tes yang akan dilakukan oleh calon asisten praktikum. Data pada penelitian ini berupa atribut hasil live koding, mata kuliah, hasil mengajar, dan kelas. Kemudian hasil klasifikasi yang didapatkan adalah sangat disarankan, disarankan, dan yang terakhir tidak disarankan. Implementasi berisi kode program dan tahapan-tahapan yang dilakukan pada hadoop. Implementasi pada permasalahan seleksi pemilihan asisten praktikum menggunakan bahasa pemrograman java. Dalam pengimplementasian menggunakan tiga *class*, satu *class* sebagai *map*, satu *class* lain sebagai *reduce*, dan *class* lainnya sebagai *class main*. Kesimpulan yang didapat adalah bahwa dengan menggunakan *hadoop multinode* dan metode *Naïve Bayes* yang digunakan untuk klasifikasi pemilihan asisten praktikum dapat diperoleh hasil yang cukup baik yaitu sebesar 80%. (Asikin, et al., 2016).

Penelitian klasifikasi dengan menggunakan algoritme AHP-*Naïve Bayes* untuk membantu pelajar SMA di dalam memilih jurusan yang tepat berdasarkan kemampuan akademiknya. Metode *Naïve Bayes* digunakan untuk mengklasifikasi data nilai pelajar ke dalam kelas fakultas. Selanjutnya metode Analytical Hierarchy Process (AHP) digunakan untuk membantu menentukan jurusan yang tepat berdasarkan jurusan di fakultas yang telah ditentukan oleh metode *Naïve Bayes*.

Dengan menggunakan metode *AHP-Naïve Bayes* ini dapat dihasilkan akurasi sistem yang mencapai 74,58% (Prasetya, et al., 2016).

Penelitian selanjutnya membuat Sistem Pendukung Keputusan Peminatan SMA Menggunakan Metode *Weighted Product* (WP). Pada penelitian ini menggunakan data berupa nilai raport Sekolah Menengah Pertama (SMP), nilai UN SMP, dan minat siswa. Data sampel untuk uji sistem diambil dari data siswa kelas X, SMA Negeri 13 Semarang. Hasil uji sistem menunjukkan 93.2% dari 103 jumlah data siswa sesuai dengan proses peminatan yang dilakukan manual di SMA N 13 Semarang (Fartindyyah & , 2014).

Penelitian selanjutnya membuat perancangan sistem untuk pemilihan lokasi perumahan menggunakan metode *Weighted Product* (WP). Sistem ini dibuat untuk mempermudah dalam pemilihan lokasi perumahan, dengan menggunakan metode *Weighted Product* (WP) dapat membantu menyelesaikan permasalahan tersebut karena pemilihan rumah dapat di golongkan ke dalam masalah yang bersifat *multiobjectives* (ada banyak tujuan yang ingin dicapai) dan *multicriteria* (ada banyak kriteria untuk mencapai tujuan) (Wijaya, 2014).

Penelitian selanjutnya dengan menggunakan *Weighted Product* untuk menentukan pilihan jurusan siswa. Untuk menentukan pilihan jurusan dengan menentukan kriteria-kriteria di setiap jurusan IPA dan IPS lalu memberikan bobot penilaian kepada beberapa kriteria yang dipertimbangkan, Hasil perhitungan akan diperingkatkan untuk mengetahui jurusan dengan nilai alternatif terbaik (Sianturi, 2013).

Penelitian selanjutnya membuat sistem untuk mempercepat dan mempermudah membuat suatu keputusan dalam pemilihan calon penerima jamkesmas, Pada penelitian ini kriteria yang digunakan sebanyak 14 kriteria menggunakan metode *Weighted Product* dengan *inputan* nilai kriteria berupa Quisioner dari pengguna dan hasil pengujian mendapatkan nilai akurasi yaitu 86,6% yang menunjukkan bahwa sistem pendukung keputusan dapat berfungsi dengan baik sesuai dengan metode *Weighted Product* (Alfinda, et al., 2014) ..

Tabel 0.1 Kajian Pustaka

No	Judul	Penulis	Perbandingan	
			Kajian Pustaka	Skripsi Penulis
1.	<i>Ranking Popular Items By Naïve Bayes Algorithm</i>	Shiramshe tty Gouthami , Golamari.Jose Mary dan	- Menggunakan Algoritme <i>Naïve Bayes</i> menyarankan barang-barang populer	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua

		Pulluri Srinivas Rao	berdasarkan <i>rangking</i>	kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data berdasarkan perhitungan alternative terbaik
2.	Implementasi Metode <i>Naïve Bayes</i> Classifier Untuk Seleksi Asisten Praktikum Pada Simulasi Hadoop Multinode Cluster	Maryamah, Moh. Fadel Asikin, Daisy Kurniawaty, Selly Kurnia Sari, Imam Cholissodin	- Menggunakan Hadoop yang dapat memproses data dalam ukuran besar dan <i>Naïve Bayes</i> untuk klasifikasi dalam pemilihan asisten praktikum	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data berdasarkan perhitungan alternative terbaik
3.	Sistem Rekomendasi Pemilihan Sekolah Menengah Tingkat Atas Menggunakan Metode <i>Naïve Bayes</i>	Triana Dewi Salma	- Menggunakan Metode <i>Naïve Bayes</i> untuk Rekomendasi Pemilihan Sekolah Menengah Tingkat Atas	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data

				berdasarkan perhitungan alternative terbaik
4.	Sistem Pengklasifikasi Jurusan Kuliah Untuk Pelajar SMA Menggunakan Algoritme Ahp- <i>Naïve Bayes</i>	Davin Ega Prasetya, Imam Cholissodin dan Agus Wahyu Widodo	- Menggunakan Algoritme <i>Naïve Bayes</i> untuk mengklasifikasi data nilai pelajar ke dalam kelas fakultas dan metode Analytical Hierarchy Process digunakan untuk membantu menentukan jurusan yang tepat berdasarkan jurusan di fakultas yang telah ditentukan oleh metode <i>Naïve Bayes</i> .	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data berdasarkan perhitungan alternative terbaik
5.	Perancangan Sistem Pendukung Keputusan Untuk Pemilihan Lokasi Perumahan Menggunakan <i>Weighted Product</i>	Rahmadi Wijaya	- Menggunakan metode <i>Weighted Product</i> untuk memilih lokasi perumahan	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data

	Method (WPM)			berdasarkan perhitungan alternative terbaik
6.	Sistem Pendukung Keputusan Untuk Menentukan Pemilihan Jurusan Siswa Dengan Menggunakan Metode <i>Weighted Product</i> (Studi Kasus: Sma Swasta Hkbp Doloksanggul)	Ingot Seen Sianturi	- Menggunakan metode <i>Weighted Product</i> untuk membantu dalam mengambil keputusan jurusan siswa berdasarkan nilai alternatif tertinggi	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data berdasarkan perhitungan alternative terbaik
7.	Development of Decision Support System to Selection of the Blended Learning Platforms for Mathematics and ICT Learning at SMK TI Udayana	I Made Ardana, I Putu Wisna Ariawan, Dewa Gede Hendra Divayana	- Menggunakan metode <i>Weighted Product</i> untuk dapat menentukan platform pembelajaran campuran untuk pembelajaran matematika dan TIK berdasarkan nilai tertinggi yang diperoleh dari perhitungan	- Menggunakan metode <i>Naïve Bayes</i> untuk menghitung probabilitas semua kemungkinan dan metode <i>Weighted Product</i> untuk melakukan penentuan kelas dari data berdasarkan perhitungan alternative terbaik

			beberapa kriteria.	
--	--	--	--------------------	--

2.2 Dasar Teori

2.2.1 Jurusan Kuliah

Memilih jurusan kuliah yang tepat akan menentukan masa depan calon Mahasiswa karena jika tidak menikmati apa yang dipelajari, maka calon Mahasiswa tersebut akan tertekan dan kemungkinan terburuk calon Mahasiswa tidak dapat berkembang di bidang tersebut padahal di jenjang perguruan tinggi merupakan modal awal untuk meniti karier. Jurusan yang tepat dapat membantu karier calon Mahasiswa tersebut untuk menuju kesuksesan di masa depan. Untuk itu, butuh mengetahui minat, bakat, dan keinginan untuk berkarier di bidang apa.

Di dalam menentukan keminatan jurusan kuliah, telah dilakukan penelitian mengenai monitoring untuk seleksi jurusan. Membantu pelajar dalam pemilihan jurusan di perguruan tinggi. Penelitian tersebut menggunakan sampel siswa Jerman dan dalam untuk tingkat penilaian menggunakan tujuan hidup dan tujuan kerja pelajar kelak (Hill, et al., 2016). Penelitian selanjutnya mengenai sistem pakar untuk membantu pelajar menentukan jurusan pendidikannya bernama Student Advisor Expert System (SAES). Penelitian tersebut dibuat karena pelajar sering membuat pilihan yang kurang tepat dan sesuai dengan minat karena kurangnya sistem petunjuk yang lengkap. Sebagian besar pelajar di India memilih jurusan berdasarkan kecenderungan di masyarakat, ketertarikan sesaat, tekanan oleh orang tua, daripada ketertarikan personal dan kemampuan. Selain itu (Deorah, et al., 2010)

Untuk memilih jurusan kuliah yang tepat, diperlukan pertimbangan matang matang, karena banyaknya jurusan kuliah yang tersedia saat ini. Keminatan jurusankuliah terbagi dalam beberapa fakultas, yang merupakan kumpulan dari beberapa jurusan dengan kesamaan disiplin ilmu. Berdasarkan daftar fakultas yang terdapat di Universitas Brawijaya, contoh fakultas di Universitas Brawijaya yaitu :

1. Fakultas Hukum
2. Fakultas Ekonomi dan Bisnis
3. Fakultas Ilmu Administrasi
4. Fakultas Pertanian
5. Fakultas Peternakan
6. Fakultas Teknik

7. Fakultas Kedokteran
8. Fakultas Perikanan dan Ilmu Kelautan
9. Fakultas Matematika dan Ilmu Pengetahuan Alam
10. Fakultas Teknologi Pertanian
11. Fakultas Ilmu Sosial dan Ilmu Politik
12. Fakultas Ilmu Budaya
13. Fakultas Kedokteran Hewan
14. Fakultas Ilmu Komputer
15. Fakultas Kedokteran Gigi
16. Program Vokasi
17. Program Pascasarjana

Masing-masing jurusan memiliki kriteria yang harus dipenuhi oleh calon Mahasiswa / pelajar. Tiap jurusan juga memiliki standar dan penilaian yang berbeda-beda untuk diujikan kepada calon Mahasiswa / pelajarnya.

2.2.2 Klasifikasi

Klasifikasi adalah suatu proses untuk menemukan suatu kelompok dari beberapa data ke dalam beberapa kelas dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Beberapa metode klasifikasi adalah *Naïve Bayes*, *Bayesian Decision* dan *Linear Discriminant Analysis* (Prasetyo, 2012)

2.2.3 Naïve Bayes

Naïve Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya. *Naïve Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai *output*. Dengan kata lain, diberikan nilai *output*, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan *Naïve Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (Training Data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. *Naïve Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan (Saleh, 2015).

Langkah-langkah yang dilakukan untuk menguji suatu data uji adalah :

1. Hitung Prior setiap kelas

$$P(C_j) = \frac{\text{Jumlah Kemunculan Data Kelas } C_j}{\text{Total Data Keseluruhan}} \quad (2.1)$$

Keterangan :

$P(C_j)$ = Nilai Prior untuk kelas C

2. Hitung Likelihood untuk setiap kriteria

Untuk menghitung likelihood terdapat 2 tipe yaitu tipe data kontinu dan tipe data diskrit. Berikut adalah rumus likelihood tipe data kontinu dan tipe data diskrit :

- Data kontinu

Menghitung Likelihood dengan tipe data kontinu digunakan densitas Gauss dengan Persamaan 2.2 :

$$P(X_i = x_i | C_j = c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (2.2)$$

Keterangan :

P = Peluang

X_i = Atribut ke i

x_i = Nilai atribut ke i

C_j = Kelas yang dicari

c_j = Sub kelas yang dicari

π = Nilai PI (22/7 atau 3.141592653589793)

e = Nilai Euler (2.718281828459045)

X_1 = Nilai Data Kontinu

μ = Nilai Mean dari data latih

σ = Nilai Standar Deviasi dari data latih

Rumus Mean :

$$\mu = \frac{\sum_{j=1}^n X_i}{n} \quad (2.3)$$

Rumus Standar Deviasi :

$$\sigma = \sqrt{\frac{(\sum_{j=1}^n X_i - \mu)^2}{n-1}} \quad (2.4)$$

Keterangan :

$\sum_{j=1}^n X_i$ = Sigma penjumlahan data x dengan indeks 1 hingga n

\bar{x} = Nilai mean dari data x

n = jumlah data

- Data Diskrit

$$P(X_i|C_j) = \frac{\text{Jumlah Kemunculan Kriteria } X_i}{\text{Total Data pada Kelas } C_j} \quad (2.5)$$

Keterangan :

$P(X_i|C_j)$ = Peluang Kondisi X_i pada kelas C_j

3. Tentukan Posterior

Menghitung posterior 1 kelas berdasarkan data uji menggunakan Persamaan 2.6:

$$P(C|X_1, X_2, X_3 \dots X_n) = P(C) \prod_{i=1}^n P(X_i | C) \quad (2.6)$$

Keterangan :

- $P(C|X_1, X_2 \dots X_n)$ = nilai posterior data dengan variabel X_1 hingga X_n untuk kelas C
- $P(C)$ = nilai prior untuk kelas C
- $\prod_{i=1}^n P(X_i | C)$ = Sigma Perkalian untuk Likelihood kondisi X dengan indeks 1 hingga n untuk kelas C

Untuk menghitung posterior dari data uji adalah dengan membandingkan nilai posterior tiap kelas. Kelas dengan nilai posterior tertinggi akan menjadi hasil klasifikasi kelas dari data uji. Detail Persamaan 2.7 adalah :

$$P(C|X_1, X_2, X_3 \dots X_n) = \arg \max P(C) \prod_{i=1}^n P(X_i | C) \quad (2.7)$$

Keterangan :

- $\arg \max$ = argument dengan nilai tertinggi untuk hasil posterior tiap kelas
- $P(C) \prod_{i=1}^n P(X_i | C_1)$ = proses perhitungan posterior untuk kelas C_i

Kelas dengan Posterior tertinggi akan menjadi hasil akhir klasifikasi dari permasalahan.

2.2.4 Weighted Product (WP)

Berdasarkan pernyataan Yoon pada penelitian di tahun 1989, metode *Weighted Product* menerapkan sistem perkalian untuk menghubungkan rating atribut, rating setiap atribut dipangkatkan terlebih dahulu dengan bobot atribut

yang bersangkutan. Tahap ini kemudian dengan normalisasi. Preferensi alternatif A_i ditunjukkan pada Persamaan 2.8 (Rudiarsih, 2012).

$$S_i = \prod_{j=1}^n x_{ij}^{w_j} \quad (2.8)$$

Keterangan :

S = Preferensi alternatif dituliskan sebagai vektor S dengan $i=1,2,\dots,m$; dimana $\sum w_j = 1$.

w_j merupakan pangkat yang bernilai positif jika termasuk atribut yang menguntungkan, sedangkan untuk atribut yang bersifat merugikan atau merupakan biaya pangkatnya bernilai negative. Preferensi relative untuk setiap alternatif ditunjukkan pada Persamaan 2.9 (Rudiarsih, 2012).

$$V_i = \frac{\prod_{j=1}^n x_{ij}^{w_j}}{\prod_{j=1}^n (x_j^*)^{w_j}} \quad (2.9)$$

Keterangan :

V = Preferensi alternatif dianalogikan sebagai vektor V

x = Nilai Kriteria

w = Bobot Kriteria / Sub Kriteria

i = Alternatif

j = Kriteria

n = Banyak kriteria

$*$ = Banyak kriteria yang telah dinilai pada vektor S

Adapun langkah langkah yang dilakukan untuk menguji suatu data uji sebagai berikut (Prasetya, et al., 2016)

4. Menentukan bobot tiap jurusan berdasarkan nilai minimum
5. Perbaikan Bobot kriteria (W) untuk mendapatkan hasil pembobotan jurusan berdasarkan perbandingan nilai calon mahasiswa dengan nilai minimum tiap jurusan dapat dilihat pada Tabel 2.2.

Tabel 0.2 Nilai Pembobotan Jurusan

JURUSAN	Mat	Fisika	Kimia	Biologi	TIK	Bind	Bing
Kedokteran Gigi	1	1	1	1	2	1	1
Gizi	1	2	3	1	3	1	3
Kedokteran	1	3	2	1	1	2	2

6. Menghitung vektor S

7. Menghitung vektor V

2.2.5 Akurasi

Akurasi merupakan nilai yang menunjukkan kesamaan antara hasil pengukuran sistem dengan hasil sebenarnya. Penghitungan tingkat akurasi dapat dilakukan pada Persamaan 2.10 (Asikin, et al., 2016).

$$Akurasi = \frac{\Sigma data\ uji\ benar}{\Sigma jumlah\ total\ data\ uji} \times 100\% \quad (2.10)$$

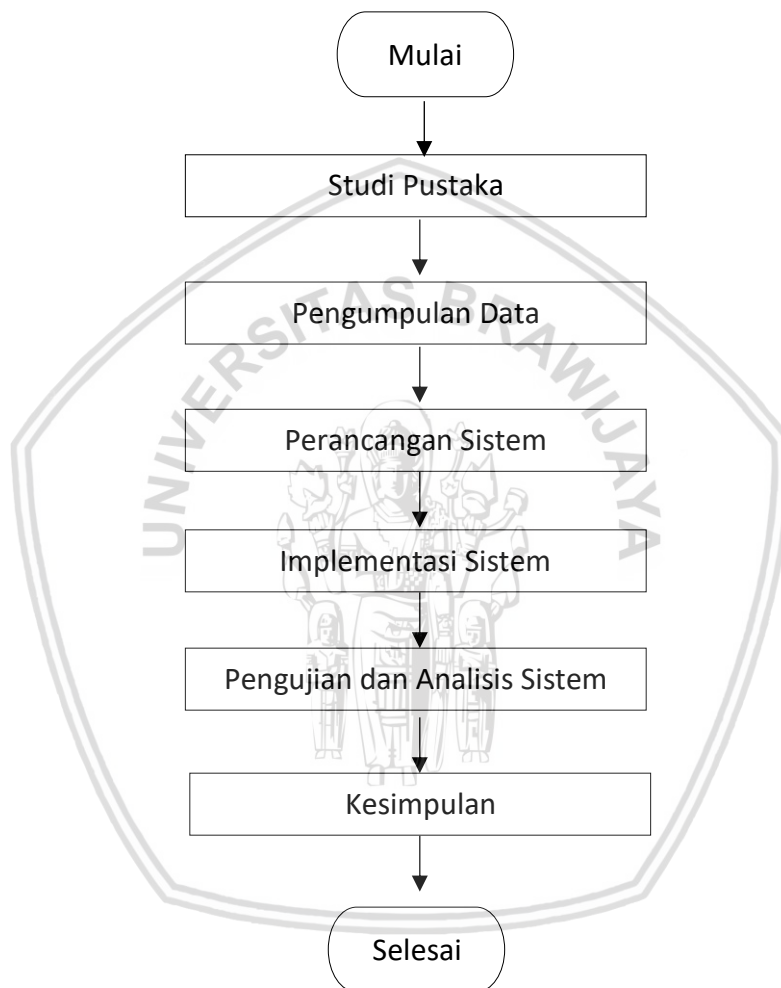
Data uji benar merupakan jumlah dari hasil pengukuran sistem yang sama dengan hasil sebenarnya. Sedangkan jumlah total data uji merupakan jumlah keseluruhan data uji yang diprediksi kelasnya.



BAB 3 METODOLOGI

3.1 Tahap Penelitian

Pada bab ini berisikan tahapan dalam penyelesaian klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA dengan menggunakan algoritme *Naïve Bayes-WP*. Untuk dapat mempermudah dalam penjelasan metodologi yang digunakan, maka penulisan menggunakan diagram alir seperti pada Gambar 3.1.



Gambar 0-1 Diagram Alir Klasifikasi Untuk Menentukan Pilihan Jurusan Kuliah Bagi Pelajar SMA dengan Menggunakan Algoritme Naïve Bayes-WP

3.2 Studi Pustaka

Dalam penelitian ini diperlukan studi literatur dari berbagai informasi dan penelitian sebelumnya yang berkaitan dengan Klasifikasi Untuk Menentukan

Pilihan Jurusan Kuliah Bagi Pelajar SMA dengan Menggunakan algoritme *Naïve Bayes-WP*.

Dasar Teori didapat dari berbagai referensi seperti artikel, buku, jurnal, internet, penelitian sebelumnya dan juga bimbingan dari dosen. Dasar teori pendukung penelitian ini yakni :

1. Keminatan Jurusan Kuliah
2. Klasifikasi
3. *Naïve Bayes*
4. *Weighted Product*

3.3 Pengumpulan Data

Pengumpulan data yang peneliti lakukan yaitu dengan cara mengambil data akademik dari pelajar di SMAN 3 Malang 2 tahun terakhir yang digunakan untuk menentukan jurusan yang cocok untuk pelajar tersebut.

3.4 Analisis Kebutuhan

Pada penelitian klasifikasi untuk menentukan pilihan jurusan kuliah bagi pelajar SMA dengan menggunakan algoritme *Naïve Bayes-WP* agar dapat diimplementasikan maka diperlukan analisis kebutuhan sistem. Sistem memerlukan spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yakni :

5. Kebutuhan perangkat keras yang diperlukan untuk implementasi sistem, yaitu :
 - Laptop atau Personal Computer (PC)
6. Kebutuhan perangkat lunak, yaitu :
 - Sistem Operasi Windows 7,8,10
 - Minimum *Netbeans* 7.2 untuk pembuatan sistem berbasis Dekstop dengan bahasa pemrograman Java.
 - Data yang akan dibutuhkan adalah data akademik pelajar SMAN 3 Malang.

3.5 Perancangan Sistem

Dalam penelitian ini sistem yang akan dibuat adalah klasifikasi untuk menentukan pilihan jurusan kuliah bagi pelajar SMA. Data yang digunakan adalah data akademik pelajar yang telah didokumentasi oleh pihak sekolah. Dalam sistem ini terdapat 2 perancangan yaitu :

7. Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna bertujuan untuk mempermudah pengguna dalam menggunakan sistem yang akan dibuat.

8. Perancangan Pengujian

Perancangan pengujian sistem yang dilakukan meliputi pengujian data latih dan data uji pada algoritme *Naïve Bayes*, pengujian data latih dan data uji pada algoritme *Naïve Bayes* dengan menggunakan data diskrit, dan pengujian bobot nilai minimum tiap jurusan pada algoritme *Weighted Product*.

3.6 Pengujian Algoritme

Pengujian pada penelitian ini dibagi menjadi dua bagian, bagian pertama yaitu pengujian terkait hasil klasifikasi data menggunakan algoritme *Naïve Bayes* untuk mengetahui fakultas apa yang terpilih dan kedua yaitu pengujian terkait hasil klasifikasi menggunakan algoritme *Weighted Product* (WP) untuk mengetahui jurusan apa yang terpilih. Skenario pengujian yang dilakukan yaitu :

9. Pengujian jumlah data latih dan data uji algoritme *Naïve Bayes*.
10. Pengujian jumlah data latih dan data uji algoritme *Naïve Bayes* dengan menggunakan data diskrit.
11. Pengujian bobot nilai minimum tiap jurusan pada algoritme *Weighted Product*.

3.6.1 Pengujian Jumlah Data Latih dan Data Uji Pada Algoritme *Naïve Bayes*

Pada pengujian ini melakukan pengujian pada jumlah data latih dan data uji pada algoritme *Naïve Bayes* yang bertujuan untuk mendapatkan jumlah data latih dan data uji yang optimal agar mendapatkan hasil klasifikasi dengan tingkat akurasi yang terbaik. Di dalam pengujian akan dilakukan pengujian dengan jumlah data latih yang berbeda, dengan jumlah sebagai berikut :

1. Pengujian dengan jumlah data latih dan data uji sebanyak jumlah data keseluruhan.
2. Pengujian dengan jumlah data latih sebanyak 90 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 10 persen dari jumlah data keseluruhan.
3. Pengujian dengan jumlah data latih sebanyak 80 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 20 persen dari jumlah data keseluruhan.
4. Pengujian dengan jumlah data latih sebanyak 70 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 30 persen dari jumlah data keseluruhan.

5. Pengujian dengan jumlah data latih sebanyak 60 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 40 persen dari jumlah data keseluruhan.
6. Pengujian dengan jumlah data latih sebanyak 50 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 50 persen dari jumlah data keseluruhan.

3.6.2 Pengujian Jumlah Data Latih dan Data Uji Pada Algoritme *Naïve Bayes* dengan Data Latih Diskrit

Pada pengujian melakukan pengujian pada jumlah data latih dan uji pada algoritme *Naïve Bayes* dengan mengubah data latih dan data uji menjadi bentuk data diskrit yang bertujuan untuk mendapatkan jumlah data latih dan data uji yang optimal agar mendapatkan hasil klasifikasi dengan tingkat akurasi yang terbaik. Di dalam pengujian akan dilakukan pengujian dengan jumlah data latih yang berbeda, dengan jumlah sebagai berikut :

1. Mengubah data latih dan data uji menjadi data diskrit.
2. Pengujian dengan jumlah data latih dan data uji sebanyak jumlah data keseluruhan.
3. Pengujian dengan jumlah data latih sebanyak 90 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 10 persen dari jumlah data keseluruhan.
4. Pengujian dengan jumlah data latih sebanyak 80 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 20 persen dari jumlah data keseluruhan.
5. Pengujian dengan jumlah data latih sebanyak 70 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 30 persen dari jumlah data keseluruhan.
6. Pengujian dengan jumlah data latih sebanyak 60 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 40 persen dari jumlah data keseluruhan.
7. Pengujian dengan jumlah data latih sebanyak 50 persen dari jumlah data keseluruhan dan jumlah data uji sebanyak 50 persen dari jumlah data keseluruhan.

3.6.3 Pengujian Bobot Nilai Minimum Tiap Jurusan pada Algoritme *Weighted Product*(WP)

Pada pengujian ini melakukan pengujian menggunakan bobot nilai minimum tiap jurusan pada algoritme *Weighted Product* ini dengan mengubah nilai bobot bertujuan untuk mendapatkan nilai bobot optimal agar mendapatkan hasil klasifikasi dan tingkat akurasi yang terbaik. Di dalam pengujian ini dilakukan

dengan menggunakan nilai bobot yang berbeda, dengan perubahan sebagai berikut :

1. Menggunakan nilai bobot dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 2 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 3 yang paling mendekati dengan nilai calon mahasiswa.
2. Menggunakan nilai bobot dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 3 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 5 yang paling mendekati dengan nilai calon mahasiswa.
3. Menggunakan nilai bobot dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 4 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 8 yang paling mendekati dengan nilai calon mahasiswa.
4. Menggunakan nilai bobot dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 5 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 9 yang paling mendekati dengan nilai calon mahasiswa.
5. Menggunakan nilai bobot dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 6 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 11 yang paling mendekati dengan nilai calon mahasiswa.

3.7 Kesimpulan

Kesimpulan pada penelitian mengenai klasifikasi untuk menentukan pilihan jurusan kuliah bagi pelajar SMA dengan menggunakan algoritme *Naïve Bayes-WP* didapat dari hasil penelitian dimulai dari studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, serta pengujian pada sistem. Kesimpulan diperoleh valid berdasarkan penelitian, sehingga menemukan tingkat akurasi dari penelitian ini.

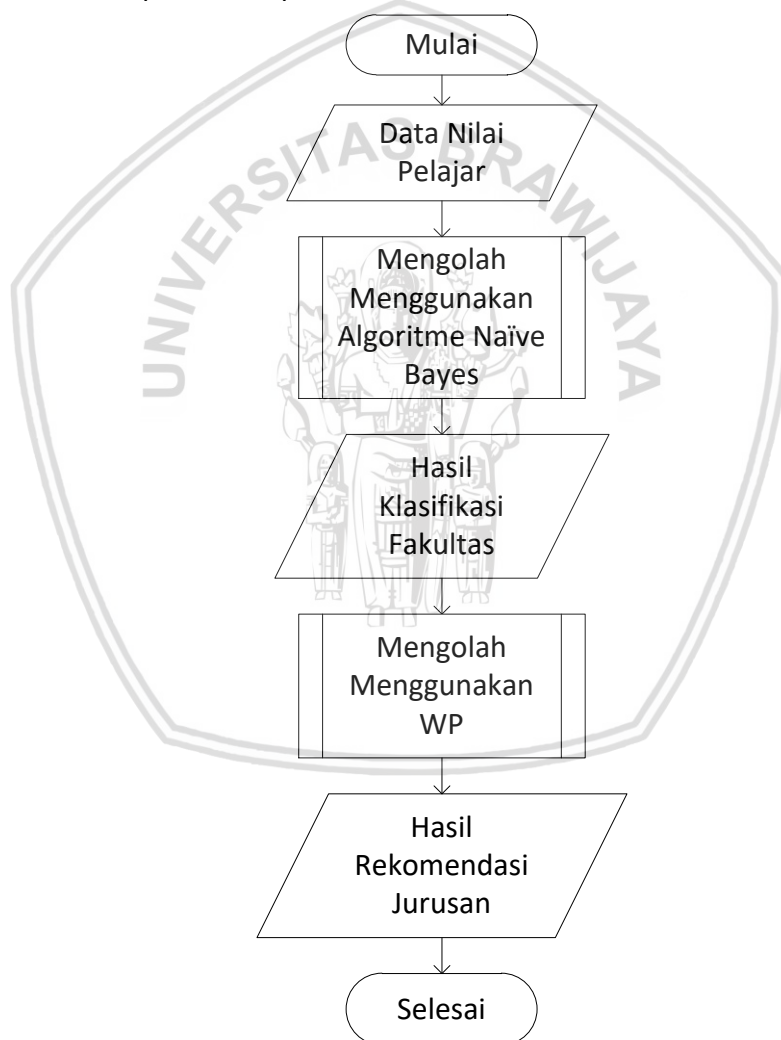
Adapula bagian saran yang berupa evaluasi mengenai kekurangan maupun kesalahan dalam penelitian sehingga dapat pula memperbaiki dan mengembangkan penelitian lebih lanjut mengenai pengklasifikasian.

BAB 4 PERANCANGAN

Pada bab perancangan ini dibahas penyelesaian masalah menggunakan algoritme *Naïve Bayes*-WP, manulisasi menggunakan algoritme *Naïve Bayes*-WP dan perancangan antar muka.

4.1 Penyelesaian Masalah Menggunakan *Naïve Bayes*-WP

Pemberian rekomendasi jurusan kuliah akan dilakukan dengan cara mengolah data latih ke dalam Algoritme *Naïve Bayes* untuk mendapatkan hasil rekomendasi fakultas dan kemudian akan diolah dengan menggunakan algoritme *WP* untuk mendapatkan hasil rekomendasi jurusan kuliah yang tepat. Rancangan *flowchart* untuk system ini dapat dilihat pada Gambar 4.1.

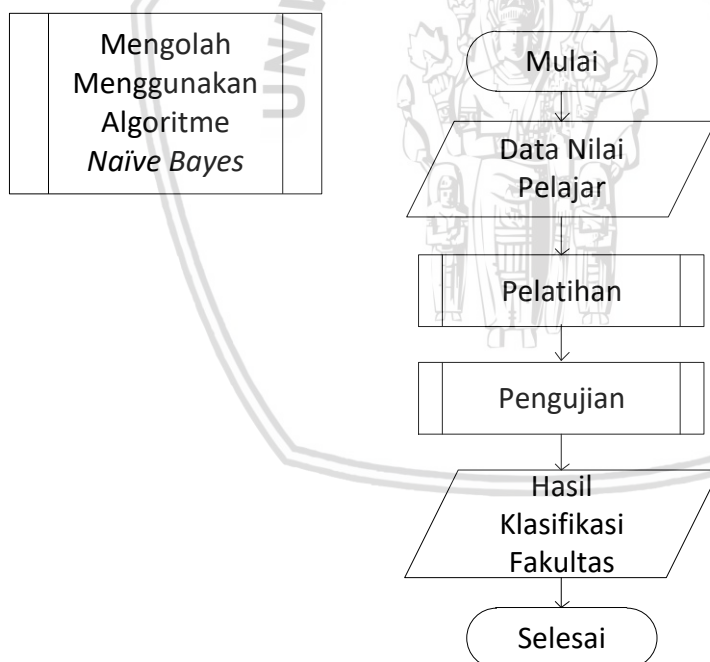


Gambar 0.1 Diagram Alir Algoritme Penyelsaian Masalah

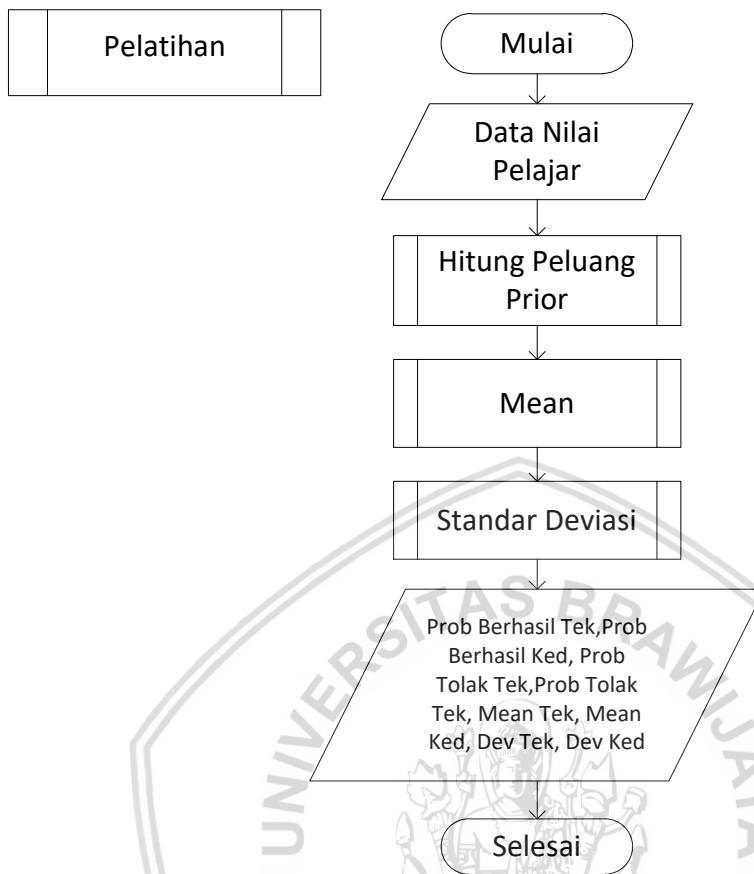
Dari Gambar 4.1 dapat dijelaskan bahwa data yang diinputkan adalah nilai akademik pelajar yang terdapat di file data *excle*. Algoritme *Naïve Bayes* berfungsi untuk mendapatkan hasil klasifikasi fakultas dari data *input*, sistem akan menampilkan *output* berupa hasil klasifikasi fakultas dan meneruskan perhitungan data uji yang memiliki hasil diterima di suatu fakultas. Setelah memiliki hasil diterima maka sistem akan mengolah data uji dengan menggunakan algoritme *Weighted Product* untuk mendapatkan rekomendasi jurusan. Sistem akan menampilkan *output* berupa rekomendasi jurusan untuk tiap data uji yang telah diinputkan berdasarkan peringkat tertinggi hingga terendah, peringkat tertinggi memiliki hasil klasifikasi jurusan yang sangat direkomendasikan.

4.1.1 Penyelesaian dengan *Naïve Bayes*

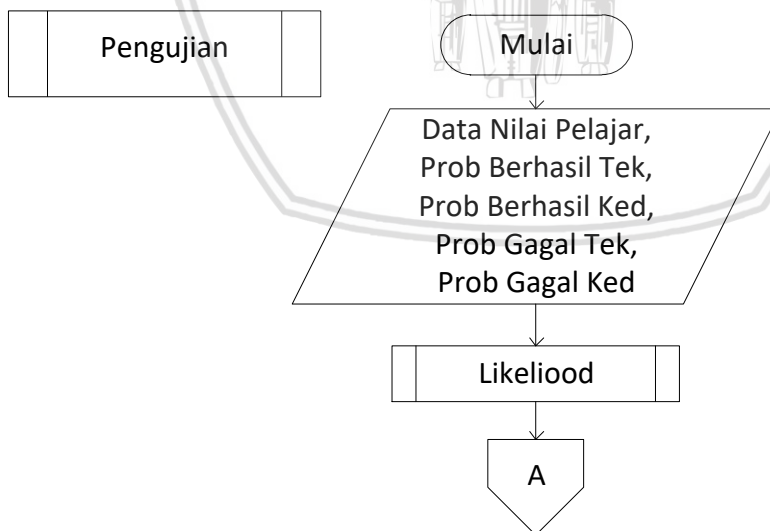
Metode *Naïve Bayes* terdiri dari perhitungan peluang prior untuk setiap kriteria selanjutnya menghitung peluang likelihood pada tiap atribut di setiap kriteria setelah mendapatkan peluang likelihood menghitung peluang posterior. Hasil posterior akan dipilih nilai yang paling tinggi untuk menentukan hasil klasifikasi kelas tersebut. Alur proses algoritme *Naïve Bayes* dapat dilihat pada Gambar 4.2 dan proses penentuan klasifikasi yang tepat pada algoritme *Naïve Bayes* dapat dilihat pada Gambar 4.5.

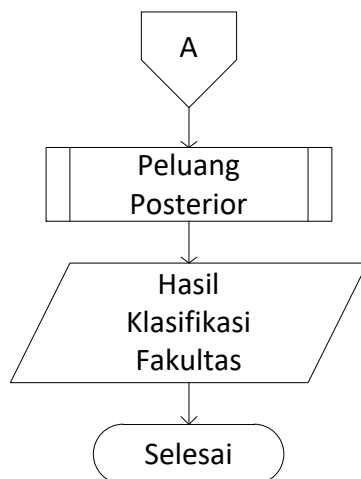


Gambar 0.2 Diagram Alir Penyelesaian *Naïve Bayes*

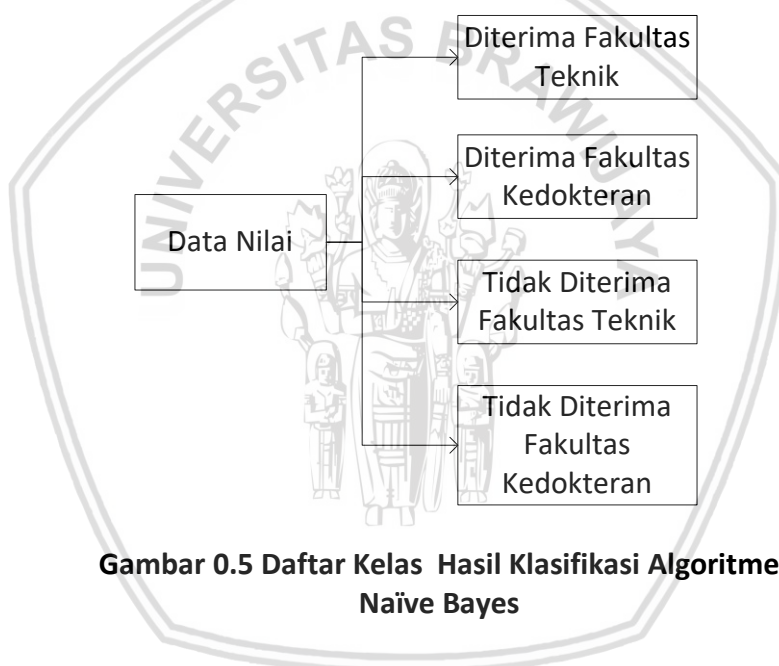


Gambar 0.3 Diagram Alir Pelatihan





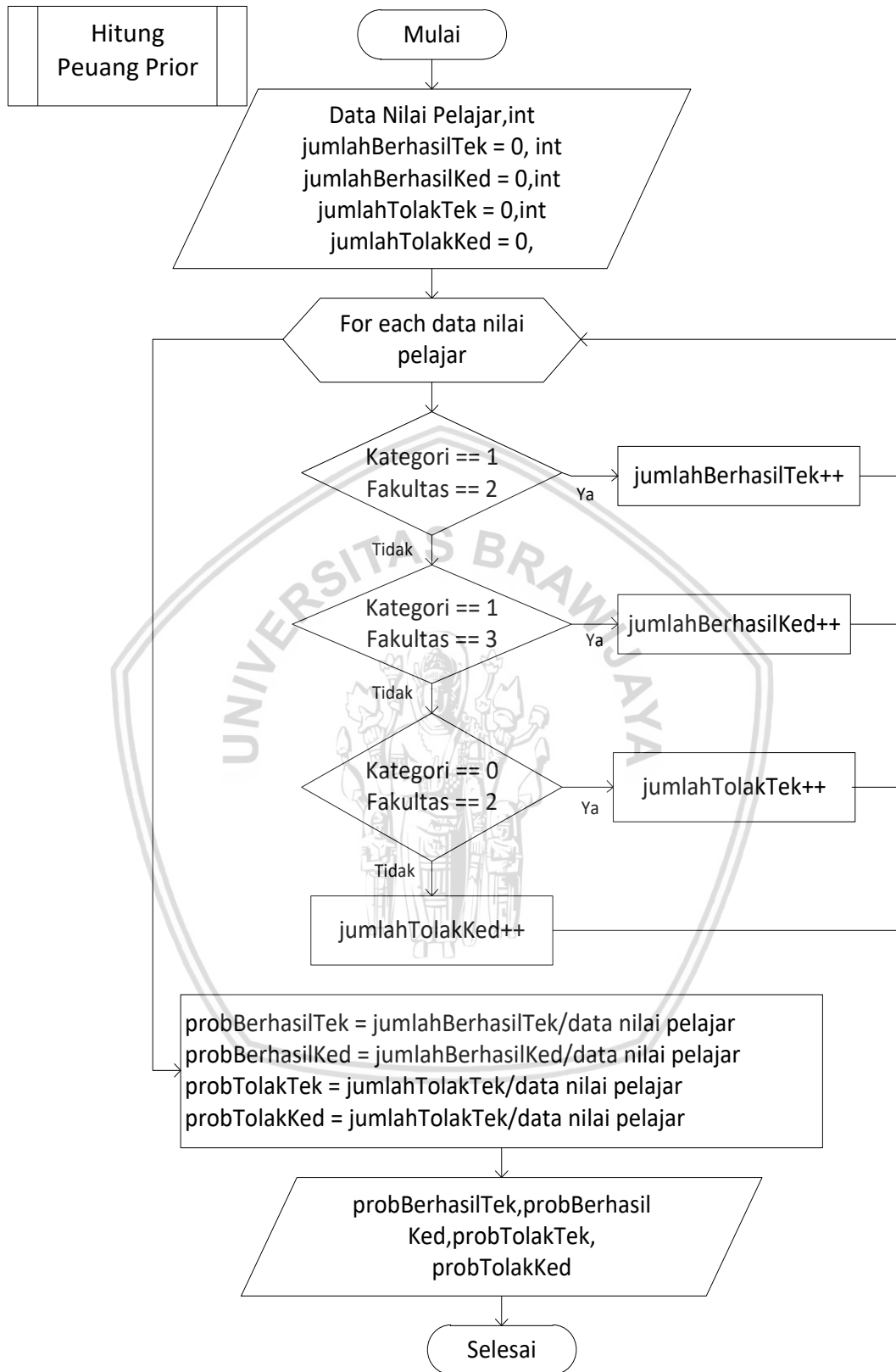
Gambar 0.4 Diagram Alir Pengujian



Gambar 0.5 Daftar Kelas Hasil Klasifikasi Algoritme Naïve Bayes

4.1.1.1 Menghitung Peluang Prior pada Tiap Kelas

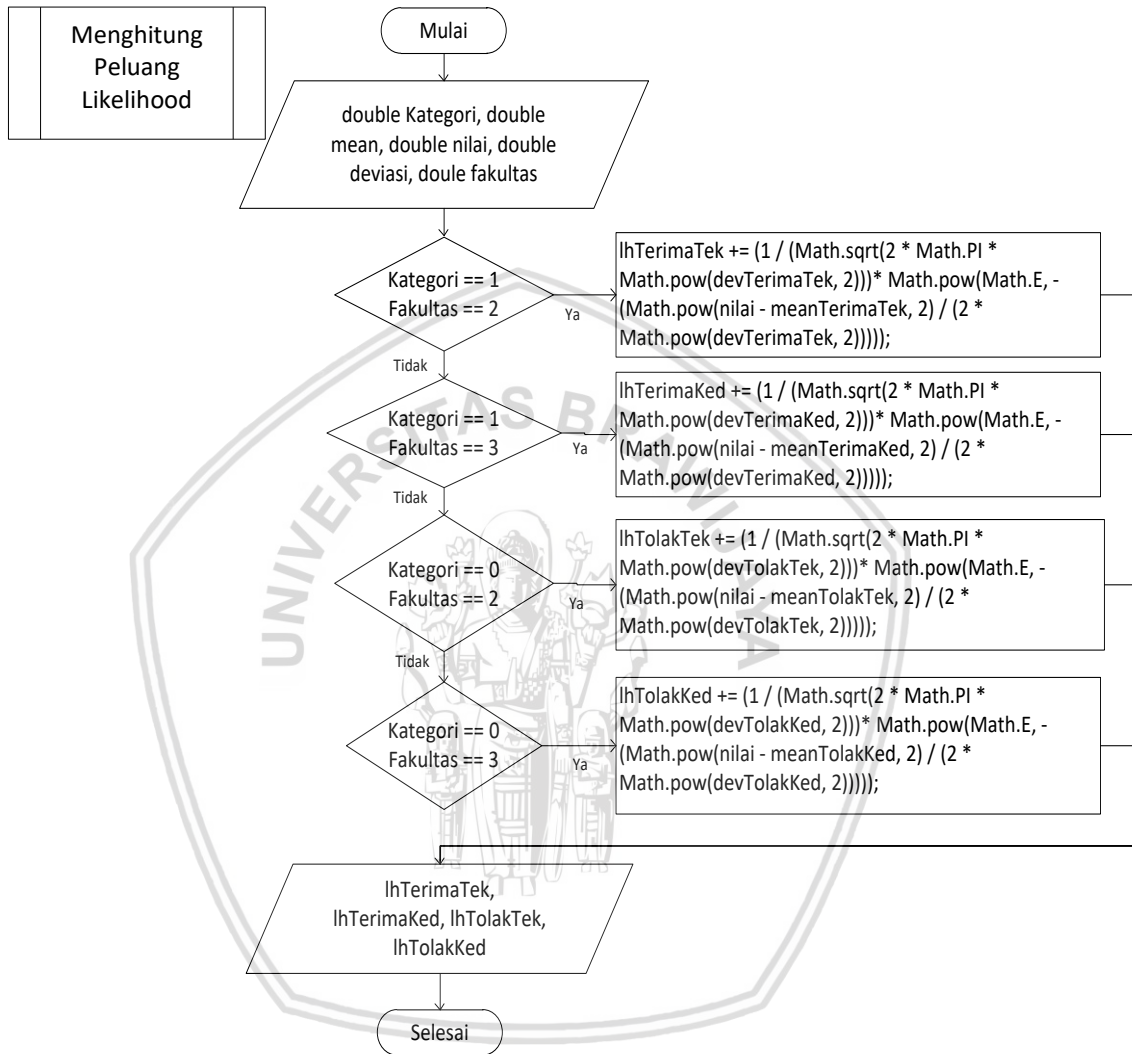
Untuk menghitung peluang prior dengan menggunakan Persamaan 2.1 dengan cara menghitung jumlah kemunculan data tiap kelas dibagi dengan total data. Penjelasan metode perhitungan likelihood dapat dilihat pada Gambar 4.6



Gambar 0.6 Diagram Perhitungan Prior

4.1.1.2 Menghitung Peluang Likelihood

Untuk menghitung likelihood menggunakan data uji. Menghitung likelihood dengan menggunakan Persamaan 2.2 dikarenakan data bertipe kontinu. Perhitungan likelihood dilakukan pada masing masing fitur. Penjelasan metode perhitungan likelihood dapat dilihat pada Gambar 4.7.

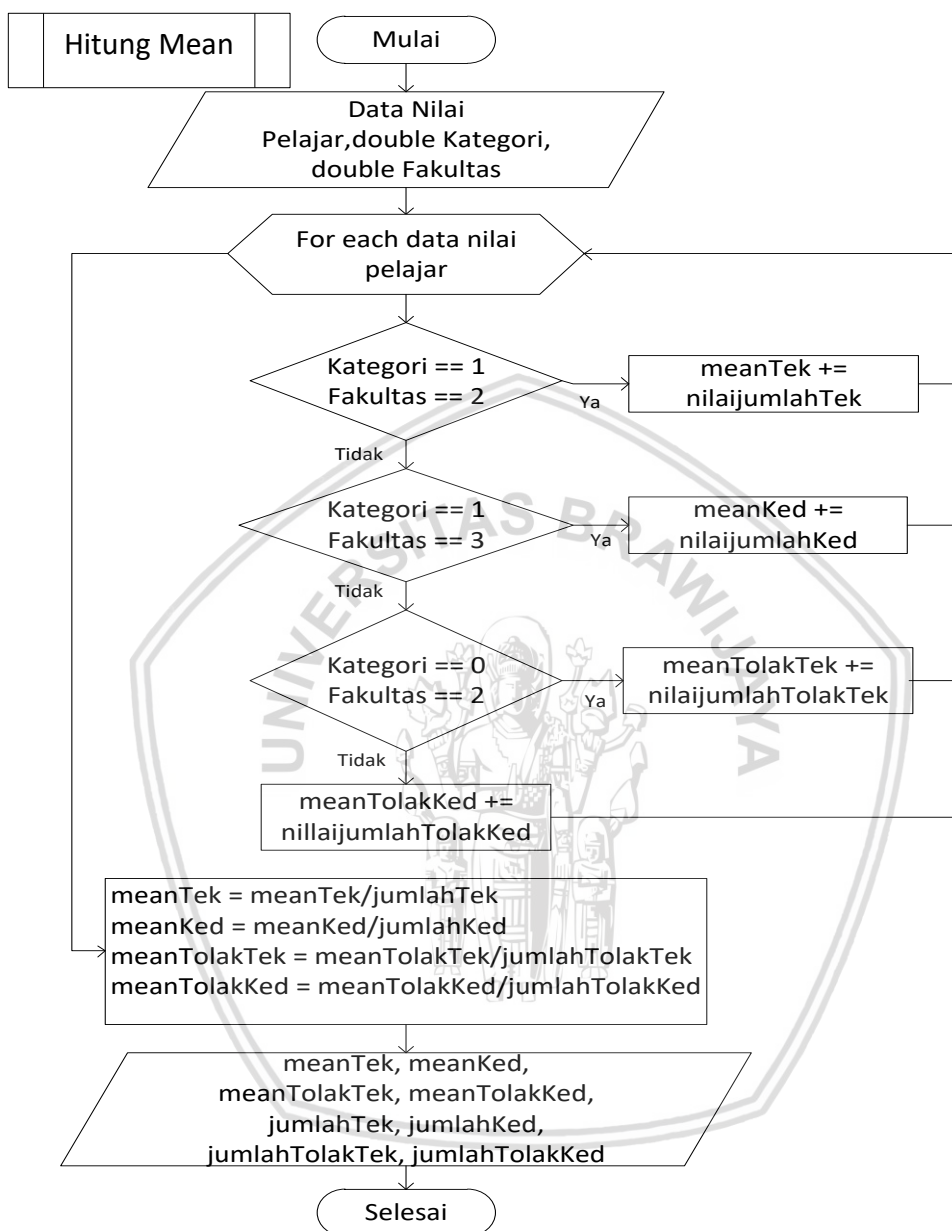


Gambar 0.7 Diagram Alir Perhitungan Likelihood

Penjelasan Diagram alir proses untuk menghitung nilai likelihood adalah sebagai berikut :

- Menghitung Nilai Mean

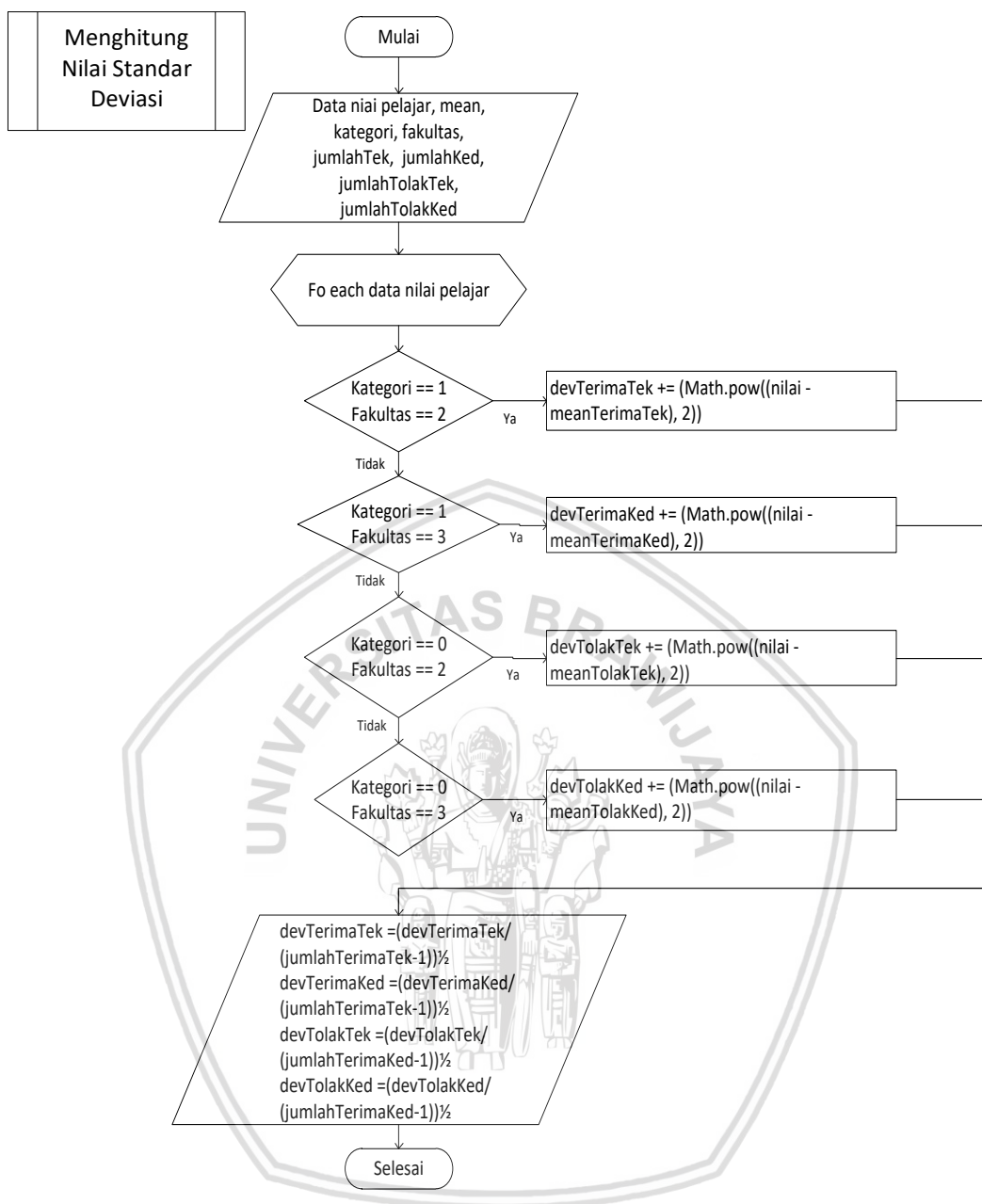
Menghitung nilai mean tiap atribut pada tiap kelas dilakukan dengan Persamaan 2.3. Untuk diagram alur proses dapat dilihat pada Gambar 4.8.



Gambar 0.8 Menghitung Mean

b. Menghitung Nilai Standar Deviasi

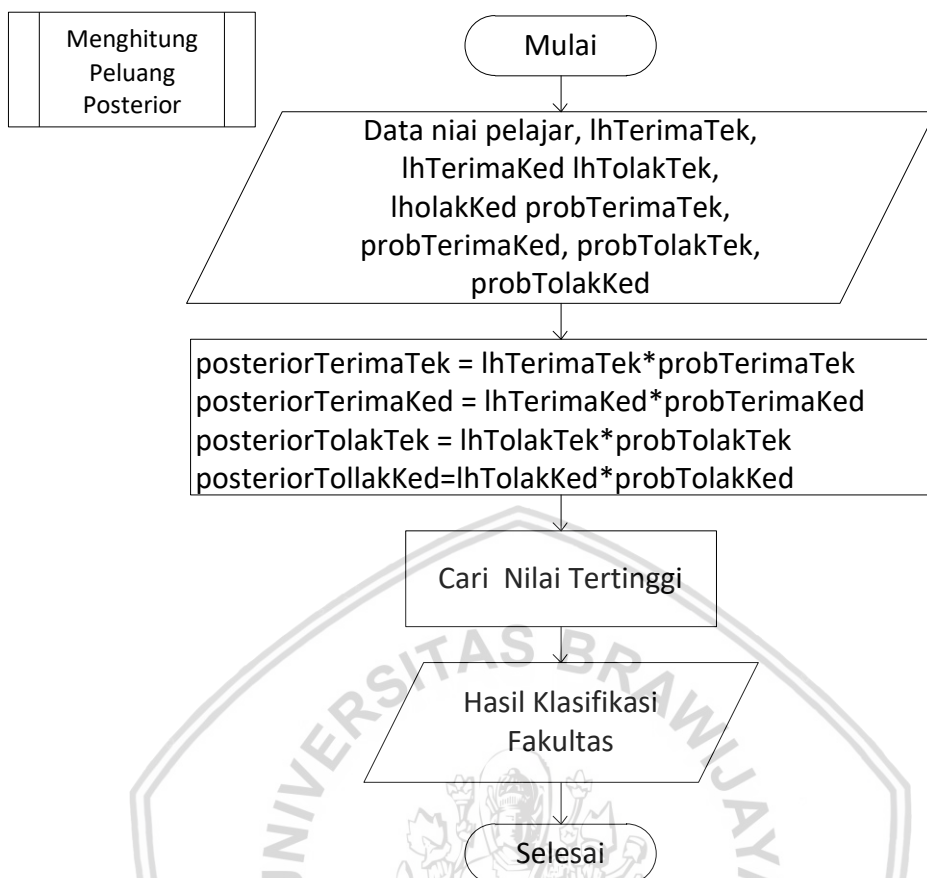
Perhitungan nilai standar deviasi tiap atribut pada tiap kelas menggunakan Persamaan 2.4 dengan hasil persamaan sebagai berikut. Untuk diagram alur proses dapat dilihat pada Gambar 4.9.



Gambar 0.9 Menghitung Standar Deviasi

4.1.1.3 Menghitung Peluang Posterior

Hasil klasifikasi ditentukan dari perbandingan posterior diterima dan ditolak dengan menggunakan rumus pada Persamaan 2.6. Posterior yang memiliki nilai terbesar akan menjadi hasil klasifikasi. Proses perhitungan posterior dapat dilihat pada Gambar 4.10.



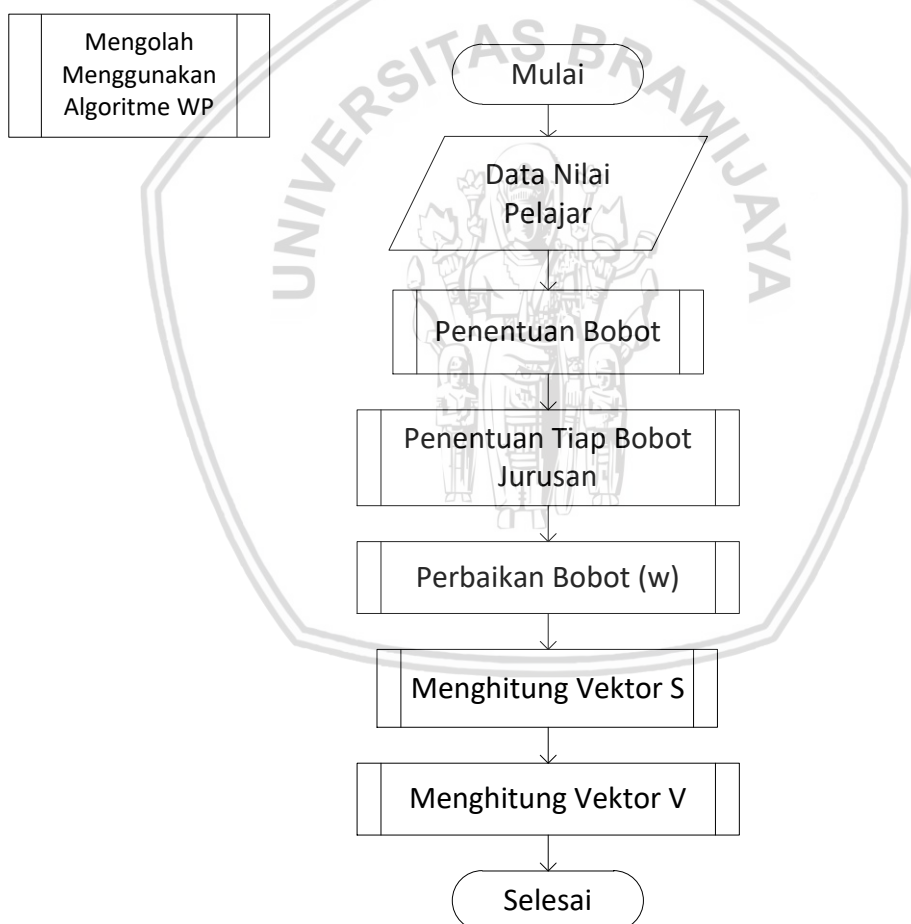
Gambar 0.10 Diagram Alir Perhitungan Posterior

4.1.2 Penyelesaian dengan *Weighted Product* (WP)

Pada bagian ini membahas tentang perancangan penelitian ini untuk memperoleh rekomendasi jurusan menggunakan algoritme WP. Langkah pertama dari bagian ini adalah mengumpulkan data akademik pelajar untuk menentukan keminatan calon mahasiswa dengan cara menghitung perbaikan bobot awal dari setiap kriteria, selanjutnya menentukan persentase nilai mata pelajaran dari calon mahasiswa. Selanjutnya, menentukan bobot dan persentase jurusan berdasarkan nilai mahasiswa. Pembobotan jurusan dilihat dari nilai minimum pada tiap jurusan dibandingkan dengan nilai calon mahasiswa. Pembobotan tersebut memiliki 5 variasi untuk pengujian untuk detailnya terdapat pada Tabel 4.1. Kemudian melakukan pembagian pada bobot serta menghitung nilai Vektor S, langkah selanjutnya menghitung Vektor V yang akan digunakan untuk penentuan keminatan jurusan. Alur proses algoritme WP dapat dilihat pada Gambar 4.11.

Tabel 0.1 Variasi bobot nilai minimum tiap jurusan

Variasi 1	Variasi 2	Variasi 3	Variasi 4	Variasi 5
Memiliki bobot bernilai 1	Memiliki bobot bernilai 1	Memiliki bobot bernilai 1	Memiliki bobot bernilai 1	Memiliki bobot bernilai 1
Memiliki bobot bernilai 2	Memiliki bobot bernilai 3	Memiliki bobot bernilai 4	Memiliki bobot bernilai 5	Memiliki bobot bernilai 6
Memiliki bobot bernilai 3	Memiliki bobot bernilai 5	Memiliki bobot bernilai 7	Memiliki bobot bernilai 9	Memiliki bobot bernilai 11



Gambar 0.11 Diagram Alir Algoritme Penyelesaian Masalah

4.1.2.1 Menentukan Bobot Tiap Mata Pelajaran

Menentukan bobot tiap mata pelajaran dengan cara yang mencari bobot tiap mata pelajaran dengan kriteria, mata pelajaran yang digunakan adalah agama, kewarganegaraan, bahasa indonesia, bahasa inggris, sejarah, seni, olahraga, fisika, matematika, kimia, dan biologi. Pembobotan dilakukan dengan menggunakan 5 kriteria, yaitu rendah dengan bobot 1 (nilai = 75 atau <75), rendah menengah dengan bobot 2 (nilai 76 - 80), menengah dengan bobot 3 (nilai 81 - 85), menengah tinggi dengan bobot 4 (nilai 86-90), dan tinggi dengan bobot 5 (nilai lebih tinggi dari 90). Aturan pembobotan dapat dilihat pada Tabel 4.2 dan keterangan Bobot dapat dilihat pada Tabel 4.3.

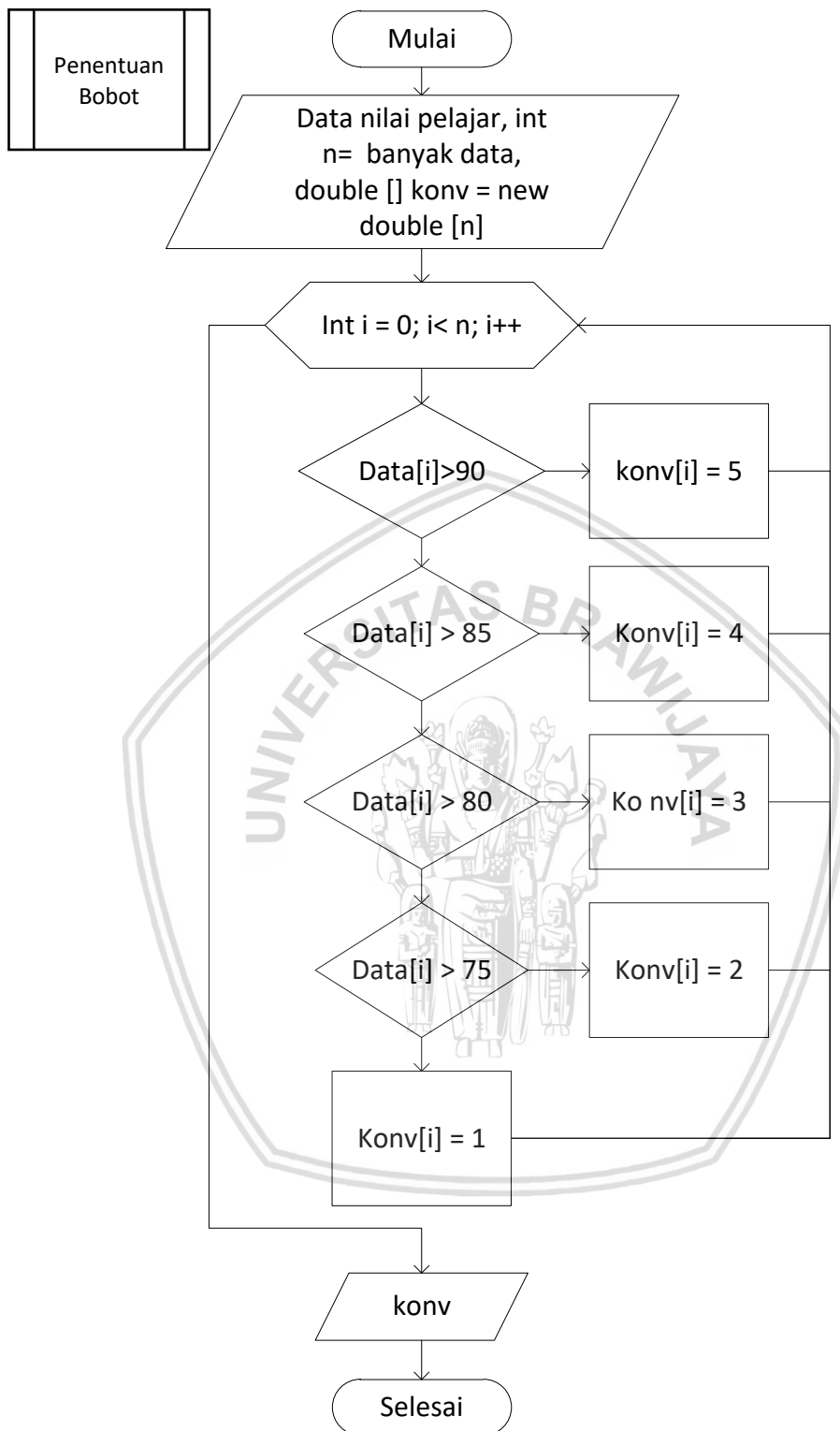
Tabel 0.2 Aturan Pembobotan

Nilai	Bobot
Nilai≤75	1
76≤Nilai≤80	2
81≤Nilai≤85	3
86≤Nilai≤90	4
Nilai>90	5

Tabel 0.3 Keterangan Bobot

Bobot	Keterangan
1	Rendah
2	Rendah Menengah
3	Menengah
4	Menengah Tinggi
5	Tinggi

Untuk diagram alir dari penentuan bobot tiap mata pelajaran dapat dilihat pada Gambar 4.12.

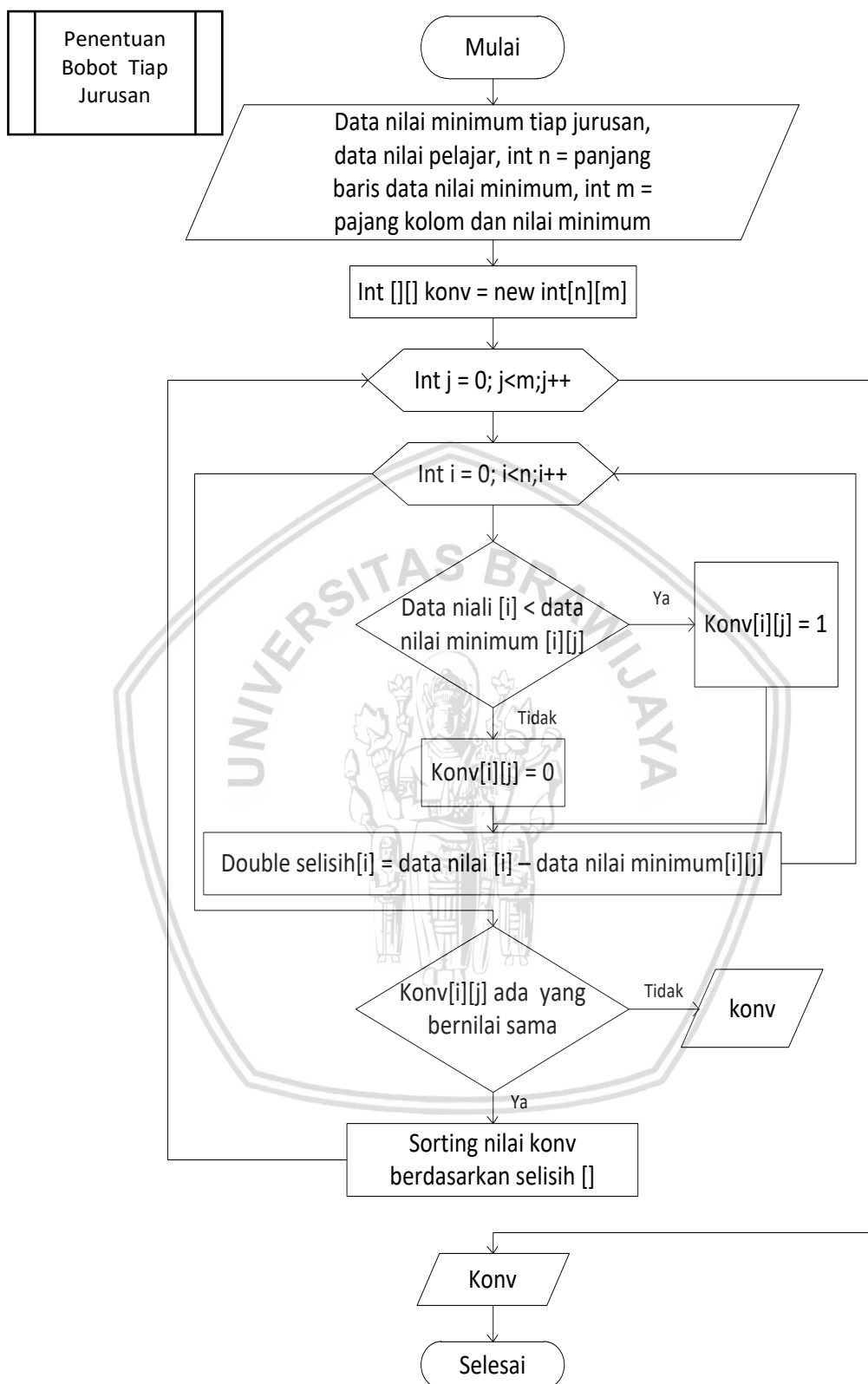


Gambar 0.12 Diagram Alir Penentuan Bobot tiap Mata Pelajaran

Dari Gambar 4.12 dapat dijelaskan bahwa data yang *diinputkan* adalah nilai akademik pelajar yang terdapat di file data. Terdapat nilai data uji pada file data dimana nilai data uji tersebut perlu diberikan nilai bobot, Untuk memberikan bobot, diberikan aturan dengan detail yang dijelaskan pada Tabel 4.1

4.1.2.2 Menentukan Bobot Tiap Jurusan Berdasarkan Nilai Minimum

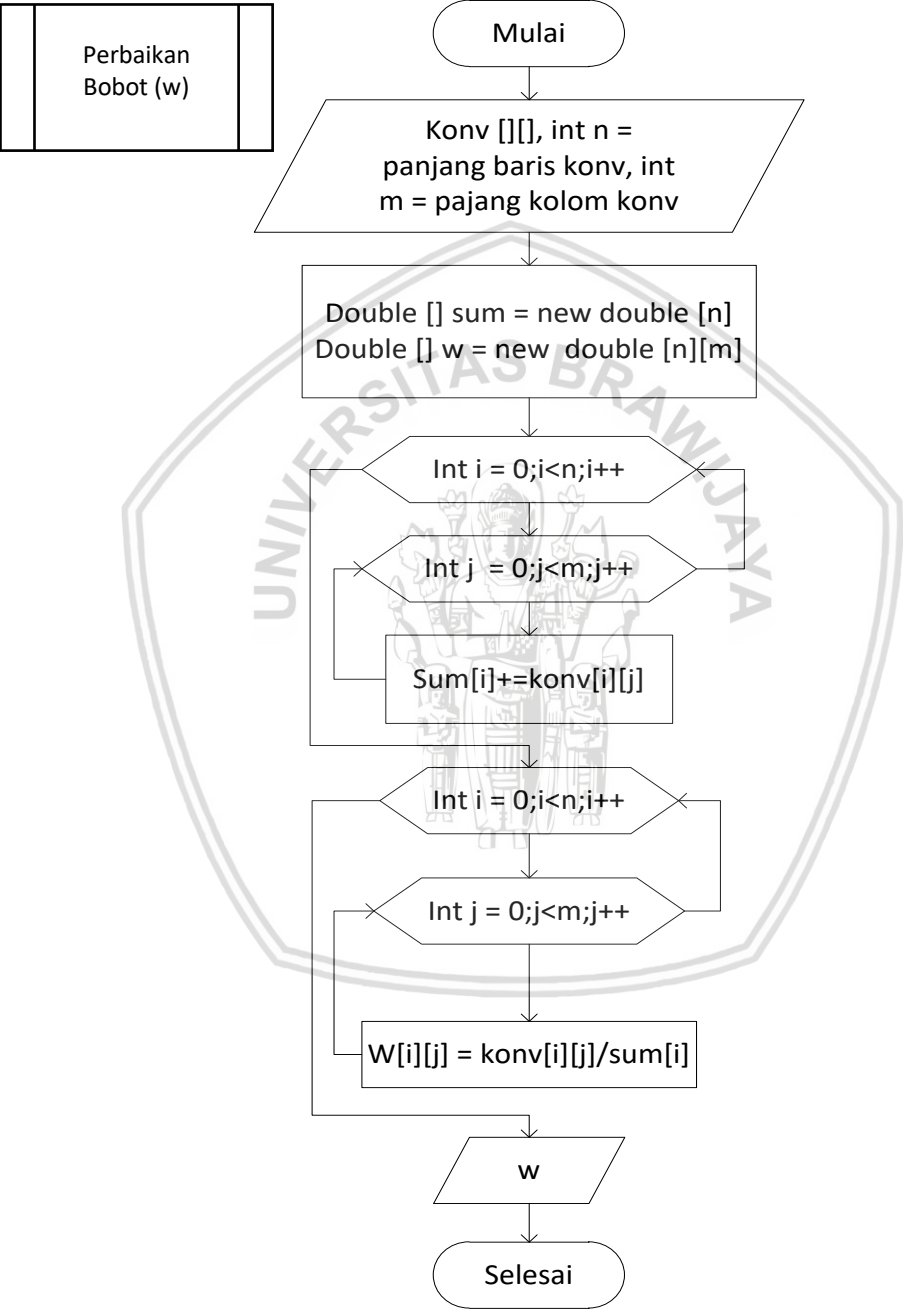
Dari Gambar 4.16 dapat dijelaskan langkah pertama yaitu mendapatkan nilai minimum tiap jurusan dengan cara membandingkan nilai dari calon mahasiswa dengan nilai minimum yang terdapat di tiap jurusan di dalam fakultas tersebut. Pada penelitian ini menggunakan 3 jurusan pada fakultas teknik yaitu teknik kimia, teknik mesin dan teknik industri dan 3 jurusan pada fakultas kedokteran yaitu dokter, farmasi, dan gizi. Setelah itu langkah kedua membandingkan nilai calon mahasiswa dengan nilai minimum fakultas, jurusan yang memiliki nilai minimum yang paling mendekati dengan nilai calon mahasiswa akan memiliki bobot tertinggi. Sebagai contoh, jika nilai calon mahasiswa adalah 83, sedangkan minimum nilai jurusan farmasi adalah 81 maka selisih bobot nilai calon mahasiswa dengan nilai minimum jurusan farmasi bernilai 2 selanjutnya minimum nilai jurusan dokter adalah 82 maka selisih bobot nilai calon mahasiswa dengan nilai minimum jurusan bernilai 1 dan minimum nilai jurusan gizi adalah 80 maka selisih bobot nilai calon mahasiswa dengan nilai minimum jurusan gizi bernilai 3, dapat dilihat bahwa selisih bobot nilai calon mahasiswa dengan nilai minimum jurusan dokter lebih tinggi dibandingkan jurusan farmasi dan gizi maka nilai calon mahasiswa lebih dekat dengan nilai minimum jurusan kedokteran. Aturan ini juga memiliki syarat yang lain yaitu jika nilai calon mahasiswa lebih rendah dari nilai minimum jurusan, maka bobot jurusan tersebut akan diberi bobot 1. Agar lebih memudahkan pemahaman dapat dilihat pada Gambar 4.13.



Gambar 0.13 Diagram Alir Penentuan Bobot Tiap Jurusan

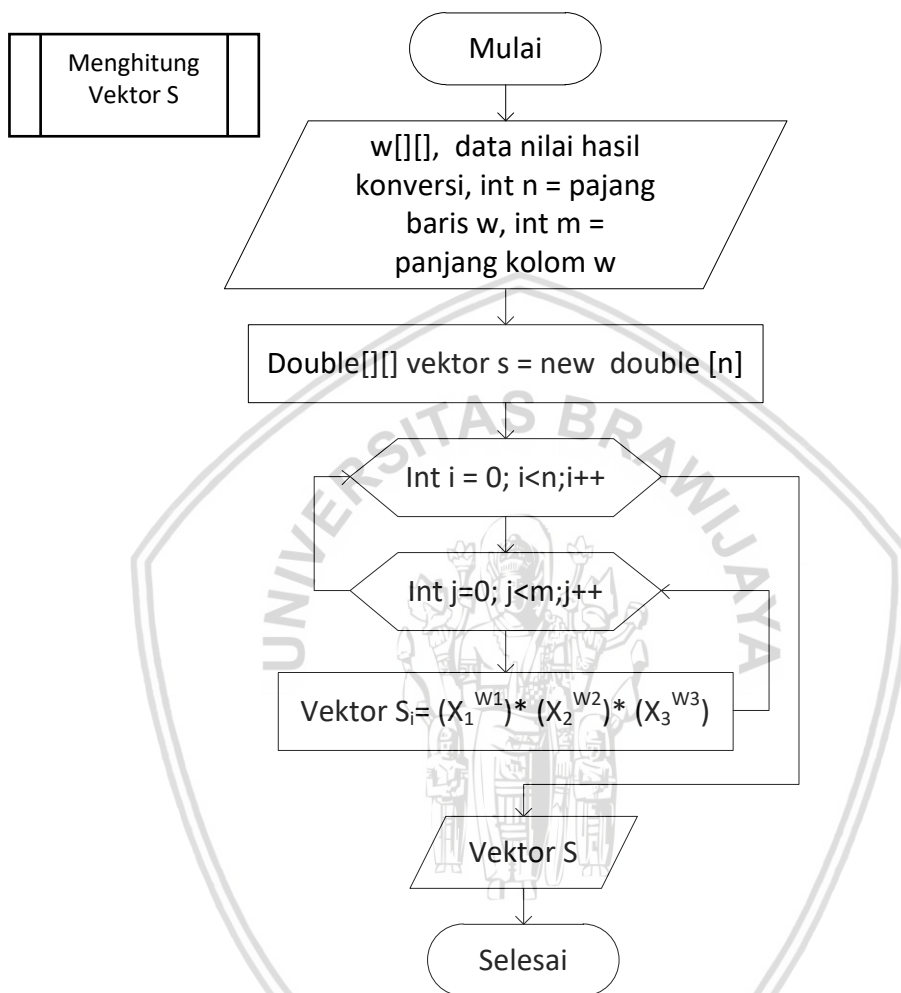
4.1.2.3 Perbaikan Bobot Kriteria (W_i)

Untuk Perbaikan Bobot Kriteria dengan cara nilai akademik calon Mahasiswa yang sudah di bobotkan pada proses membandingkan nilai calon mahasiswa denga nilai minimum jurusan yang selanjutnya nilai tersebut di bagi dengan seluruh jumlah nilai bobot pada tiap jurusan. Agar lebih memudahkan pemahaman dapat dilihat pada Gambar 4.14.



4.1.2.3 Menghitung Vektor S

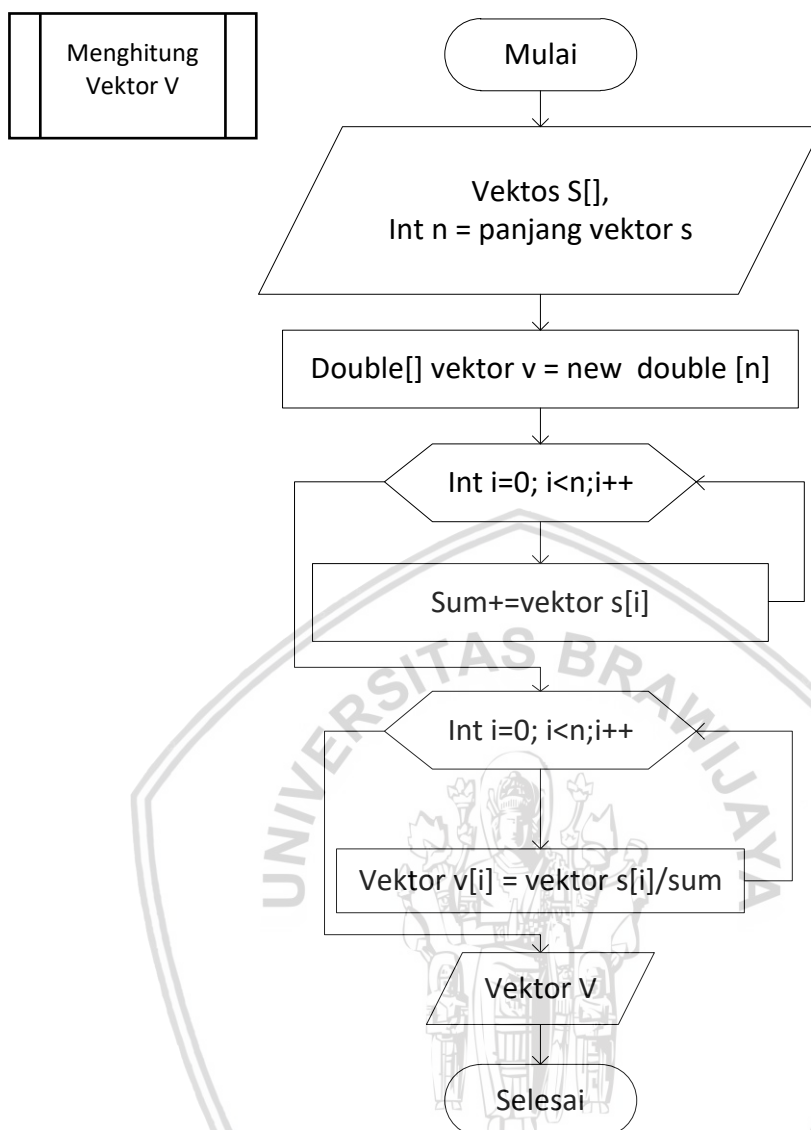
Untuk menghitung vektor S dengan cara nilai akademik calon mahasiswa yang sudah diberi bobot akan menjadi nilai pangkat dari nilai perbaikan bobot kriteria (W) yang sudah dicari sebelumnya. Agar lebih memudahkan pemahaman dapat dilihat pada Gambar 4.15.



Gambar 0.15 Diagram Alir Menghitung Vektor S

4.1.2.3 Menghitung Vektor V

Untuk menghitung vektor V dengan cara nilai vektor S yang sebelumnya sudah di cari pada tiap jurusan dibagi dengan jumlah seluruh nilai vektor S. Agar lebih memudahkan pemahaman dapat dilihat pada Gambar 4.16.



Gambar 0.16 Diagram Alir Menghitung Vektor (V_i)

4.2 Manualisasi Menggunakan Metode *Naïve Bayes*-WP

4.2.1 Manualisasi *Naïve Bayes*

4.2.1.1 *Input Data Latih*

Data latih yang digunakan pada perhitungan memiliki atribut berupa nilai akademik dari calon mahasiswa. Pada perhitungan ini digunakan data *sample* berupa data akademik dari calon mahasiswa yang berjumlah 10 orang. Data dapat dilihat pada Tabel 4.4.

Tabel 0.4 Data Uji Sample

Agama	Kwn	Indonesia	Inggris	Sejarah	Seni	Olahraga	Fisika	Matematika	Kimia	Biologi
86	89,33	88,67	90,67	89	86	96	88,67	89,33	89,67	91,67
89,33	89,67	83,7	90	87,67	88,67	90	92	90,67	92,33	93,67
95,33	95,33	89	88	93	91,33	86,67	87,67	94	94	92,33
88	90,33	84,33	93,33	89,67	97,67	87,67	94	93,33	96,67	92
92,33	91,33	89	92	97,67	89	87,33	94,33	91	96,33	96
81,33	83,33	84,67	86,33	79	82,67	85,33	78	79,67	78,67	85,67
88,33	89	82	90,33	87	84	86,67	67	86,33	86	92,67
90,33	89,33	82,33	90	86,67	93,33	86,33	89,67	81,67	85,33	88,33
87,33	89,67	88,33	88,67	89,33	84,67	85,33	81,67	82	86	84,33
83,33	89,67	87	87,67	80	89	86	81,33	82,33	84	80

4.2.1.2 Menghitung Peluang Prior pada Tiap Kelas

Pada perhitungan *Naïve Bayes* data latih prior dilambangkan dengan $P(\text{kondisi})$. Pada perhitungan ini terdapat 2 peluang, yaitu diterima atau ditolak, sehingga memiliki 2 prior yaitu $P(\text{diterima})$ dan $P(\text{ditolak})$. Langkah untuk menghitung peluang prior sebagai berikut :

2. Menghitung total seluruh data latih. Untuk perhitungan ini menggunakan data latih sample yang berjumlah 10
3. Menghitung data tiap kelas di dalam data latih. Untuk perhitungan ini memiliki 2 kelas yaitu diterima dengan jumlah 5 dan ditolak dengan jumlah 5.
4. Menghitung Prior pada data latih yang memiliki 2 kelas yaitu diterima dan ditolak menggunakan Persamaan 2.1. Dengan perhitungan sebagai berikut:
 - Prior Diterima

$$P(\text{diterima}) = \frac{5}{10}$$

$$P(\text{diterima}) = 0,5$$
 - Prior Ditolak

$$P(\text{ditolak}) = \frac{5}{10}$$

$$P(\text{ditolak}) = 0,5$$

4.2.1.3 Input Data Uji

Data uji yang digunakan pada perhitungan memiliki atribut berupa nilai akademik dari calon mahasiswa. Pada perhitungan ini digunakan data uji *sample*

berupa data akademik dari calon mahasiswa yang berjumlah 1 orang. Data dapat dilihat pada Tabel 4.5.

Tabel 0.5 Data Uji Sample

Agama	Kwn	Indonesia	Inggris	Sejarah	Seni	Olahraga	Fisika	Matematika	Kimia	Biologi
89,33	89,67	83,67	90	87,67	88,67	90	92	90,67	92,33	93,67

4.2.1.4 Menghitung Peluang Likelihood

Menghitung peluang likelihood menggunakan data uji, pada perhitungan ini menggunakan Persamaan 2.2 dikarenakan data bertipe kontinu. Perhitungan likelihood dilakukan pada masing masing atribut. Contoh perhitungan untuk likelihood menggunakan data peluang matapelajar agama untuk yang diterima $P(X=\text{Nilai Agama}|\text{diterima})$. Langkah perhitungan sebagai berikut:

5. Menghitung nilai mean menggunakan Persamaan 2.3 dengan variabel = agama dan kelas = diterima. Dengan contoh perhitungan sebagai berikut :
 - $\mu = (\text{Nilai Agama Data 1} + \text{Nilai Agama Data 2} + \dots + \text{Nilai Agama Data } n)/5$
 - $\mu = (86+89,33+95,33+88+92,33)/5 = \mathbf{90,198}$
6. Menghitung nilai standar deviasi menggunakan Persamaan 2.4 dengan variabel = agama dan kelas = diterima. Dengan contoh perhitungan sebagai berikut :
 - $\sigma = \sqrt{((\text{data agama 1} - \text{mean agama})^2 + (\text{data agama 2} - \text{mean agama})^2 + (\text{data agama 3} - \text{mean agama})^2 + (\text{data agama 4} - \text{mean agama})^2 + (\text{data agama 5} - \text{mean agama})^2)/5}$
 - $\sigma = \sqrt{((86-90,198)^2 + (89,33-90,198)^2 + (95,33-90,198)^2 + (88-90,198)^2 + (92,33-90,198)^2)/5}$
 - $\sigma = \mathbf{3,677}$
7. Menghitung likelihood menggunakan Persamaan 2.2 dengan data = 93,33 dan kelas = diterima. Dengan contoh perhitungan sebagai berikut :
 - $P(89,33|\text{diterima}) = \frac{1}{\sqrt{2*\pi*(3,677)^2}} e^{-\frac{(89,33-90,198)^2}{2*(3,677)^2}}$
 - $P(89,33|\text{diterima}) = \mathbf{0,105507}$

Hasil perhitungan likelihood pada masing-masing atribut dalam kelas secara lengkap dapat dilihat pada Tabel 4.6 dan Tabel 4.7.

- Likelihood diterima

Tabel 0.6 Likelihood Atribut untuk Peluang Diterima

Likelihood	Hasil Peluang
$P(\text{Agama}=89,33 \text{Diterima})$	0,105507

$P(\text{Kwn} = 89,67 \text{Diterima})$	0,134640
$P(\text{Indonesia} = 83,67 \text{Diterima})$	0,071055
$P(\text{Inggris} = 90 \text{Diterima})$	0,182523
$P(\text{Sejarah} = 87,67 \text{Diterima})$	0,064504
$P(\text{Seni} = 88,67 \text{Diterima})$	0,082668
$P(\text{Olahraga} = 90 \text{Diterima})$	0,103494
$P(\text{Fisika} = 92 \text{Diterima})$	0,127983
$P(\text{Matematika} = 90,67 \text{Diterima})$	0,180009
$P(\text{Kimia} = 92,33 \text{Diterima})$	0,120641
$P(\text{Biologi} = 93,67 \text{Diterima})$	0,214937

- Likelihood ditolak :

Tabel 0.7 Likelihood Atribut untuk Peluang Ditolak

Likelihood	Hasil Peluang
$P(\text{Agama} = 89,33 \text{Ditolak})$	0,074173
$P(\text{Kwn} = 89,67 \text{Ditolak})$	0,126195
$P(\text{Indonesia} = 83,67 \text{Ditolak})$	0,130266
$P(\text{Inggris} = 90 \text{Ditolak})$	0,168506
$P(\text{Sejarah} = 87,67 \text{Ditolak})$	0,067344
$P(\text{Seni} = 88,67 \text{Ditolak})$	0,082539
$P(\text{Olahraga} = 90 \text{Ditolak})$	6,16E-11
$P(\text{Fisika} = 92 \text{Ditolak})$	0,015357
$P(\text{Matematika} = 90,67 \text{Ditolak})$	0,000499
$P(\text{Kimia} = 92,33 \text{Ditolak})$	0,003407
$P(\text{Biologi} = 93,67 \text{Ditolak})$	0,024057

4.2.1.5 Menghitung Peluang Posterior

Menghitung peluang posterior pada tiap kelas digunakan untuk menentukan klasifikasi dari perbandingan peluang posterior diterima dan peluang posterior ditolak. Peluang posterior yang memiliki nilai tinggi akan menjadi hasil klasifikasi. Untuk contoh menghitung peluang posterior sebagai berikut :

8. Menghitung nilai posterior dari tiap kelas dengan menggunakan Persamaan 2.6

- Posterior diterima :

$$= P(\text{Diterima}) * P(\text{Agama} = 89,33 | \text{Diterima}) * P(\text{Kwn} = 89,67 | \text{Diterima}) * P(\text{Indonesia} = 83,67 | \text{Diterima}) * P(\text{Inggris} = 90 | \text{Diterima}) * P(\text{Sejarah} = 87,67 | \text{Diterima}) * P(\text{Seni} = 88,67 | \text{Diterima}) * P(\text{Olahraga} = 90 | \text{Diterima}) * P(\text{Fisika} = 92 | \text{Diterima}) * P(\text{Matematika} = 90,67 | \text{Diterima}) * P(\text{Kimia} = 92,33 | \text{Diterima}) * P(\text{Biologi} = 93,67 | \text{Diterima})$$

$$= \mathbf{3,03686E-11}$$

- Posterior ditolak

$$= P(\text{Ditolak}) * P(\text{Agama} = 89,33 | \text{Ditolak}) * P(\text{Kwn} = 89,67 | \text{Ditolak}) * P(\text{Indonesia} = 83,67 | \text{Ditolak}) * P(\text{Inggris} = 90 | \text{Ditolak}) * P(\text{Sejarah} = 87,67 | \text{Ditolak}) * P(\text{Seni} = 88,67 | \text{Ditolak}) * P(\text{Olahraga} = 90 | \text{Ditolak}) * P(\text{Fisika} = 92 | \text{Ditolak}) * P(\text{Matematika} = 90,67 | \text{Ditolak}) * P(\text{Kimia} = 92,33 | \text{Ditolak}) * P(\text{Biologi} = 93,67 | \text{Ditolak})$$

$$= \mathbf{2,20816E-26}$$

9. Membandingkan nilai peluang posterior dari tiap kelas untuk mencari nilai tertinggi dengan menggunakan Persamaan 2.7 menghasilkan nilai $\mathbf{3,03686E-11}$

4.2.2 Manualisasi *Weighted Product* (WP)

4.2.2.1 Menentukan Bobot Tiap Mata Pelajaran

Menentukan bobot tiap mata pelajaran dengan cara yang mencari bobot tiap mata pelajaran dengan kriteria. Pembobotan dilakukan dengan menggunakan 5 kriteria yang terdapat pada Tabel 4.1 untuk aturan pembobotan dan Tabel 4.2 keterangan bobot. Untuk menentukan bobot dapat dilakukan dengan langkah berikut :

1. *Input* data uji sampel yang berasal dari data akademik dari calon mahasiswa. Data uji sampel yang digunakan pada perhitungan memiliki atribut berupa nilai akademik dari calon mahasiswa. Pada perhitungan ini digunakan data akademik sample berupa data akademik dari calon mahasiswa yang berjumlah 1 orang. Data dapat dilihat pada Tabel 4.8.

Tabel 0.8 Data Uji Sampel

Agama	Kwn	Indonesia	Inggris	Sejarah	Seni	Olahraga	Fisika	Matematika	Kimia	Biologi
89,33	89,67	83,67	90	87,67	88,67	90	92	90,67	92,33	93,67

2. Mencocokkan data akademik per mata pelajaran dengan aturan pembobotan yang terdapat pada Tabel 4.1.
3. Memberi nilai bobot dengan aturan yang sesuai pada Tabel 4.1 setelah itu hasil pembobotan dapat dilihat pada Tabel 4.9 dan keterangan bobot dapat dilihat pada Tabel 4.2 .

Tabel 0.9 Bobot Data Uji Sampel

Agama	Kwn	Indonesia	Inggris	Sejarah	Seni	Olahraga	Fisika	Matematika	Kimia	Biologi
89,33	89,67	83,67	90	87,67	88,67	90	92	90,67	92,33	93,67
4	4	3	4	4	4	4	5	5	5	5

4.2.2.2 Menentukan Bobot Tiap Jurusan Berdasarkan Nilai Minimum

Menentukan bobot pada tiap jurusan berdasarkan nilai minimum jurusan dilakukan dengan membandingkan nilai dari calon mahasiswa dengan nilai minimum yang terdapat pada tiap jurusan di dalam fakultas tersebut. Pada perbandingan ini menggunakan 3 jurusan sampel, yaitu jurusan teknik mesin, teknik industri dan teknik kimia. Untuk menentukan bobot tiap jurusan memiliki langkah sebagai berikut:

1. Mencari nilai minimum tiap jurusan dapat dilihat pada Tabel 4.10.

Tabel 0.10 Nilai Minimum tiap Jurusan

Jurusan	Agm	Kwn	Ind	Ing	Sjr	Seni	Or	Fis	Mat	Kim	Bio
Mesin	92,33	91,33	89	88	93	89	86,67	87,67	91	94	92,33
Industri	86	89,33	88,67	90,67	89	86	96	88,67	89,33	89,67	91,67
Kimia	88	89,67	83,67	90	87,67	88,67	87,67	92	90,67	92,33	92

2. Membandingkan nilai calon mahasiswa dengan nilai minimum jurusan. Nilai minimum yang paling mendekati dengan nilai calon mahasiswa akan diberi bobot tertinggi dan bila nilai calon mahasiswa kurang dari nilai minimum akan diberi bobot 1. Hasil pembobotan jurusan berdasarkan perbandingan nilai calon mahasiswa dengan nilai minimum tiap jurusan dapat dilihat pada Tabel 4.11.

Tabel 0.11 Bobot Perbandingan Nilai Calon Mahasiswa dengan Nilai Minimum Jurusan

Jurusan	Agm	Kwn	Ind	Ing	Sje	Seni	Or	Fis	Mat	Kim	Bio
Mesin	1	1	1	1	1	1	2	1	1	1	3
Industri	2	2	1	1	1	2	1	2	2	2	1
Kimia	3	3	2	2	2	3	3	3	3	3	2

4.2.2.3 Perbaikan Bobot Kriteria (W_i)

Perbaikan Bobot Kriteria dengan cara nilai akademik calon Mahasiswa yang sudah di bobotkan pada proses membandingkan nilai calon mahasiswa dengan nilai minimum jurusan yang selanjutnya nilai tersebut di bagi dengan seluruh jumlah nilai bobot pada tiap jurusan. Langkah langkah perhitungan perbagikan bobot kriteria sebagai berikut:

1. Menghitung total seluruh bobot kriteria per jurusan yang didapat hasil
 - Total Bobot Kriteria Jurusan Mesin = 14
 - Total Bobot Kriteria Jurusan Industri = 17
 - Total Bobot Kriteria Jurusan Kimia = 29
2. Mencari nilai W dengan setiap bobot kriteria perjurusan dibagi dengan total bobot kriteria per jurusan. Dengan contoh perhitungan sebagai berikut :

$$\begin{aligned}
 W &= \frac{\text{Bobot Kriteria Jurusan per Mata Pelajaran}}{\text{Total Bobot Kriteria Jurusan}} \\
 W &= \frac{\text{Bobot Kriteria Jurusan Mesin pada Mata Pelajaran Agama}}{\text{Total Bobot Kriteria Jurusan Mesin}} \\
 W &= \frac{1}{13} \\
 W &= 0,0769
 \end{aligned}$$

Hasil perhitungan nilai W pada masing-masing jurusan per mata pelajaran secara lengkap dapat dilihat pada Tabel 4.12.

Tabel 0.12 Perbaikan Bobot Kriteria(W_i)

Jurusan	Agm	Kwn	Ind	Ing	Sjr	Seni	Or	Fis	Mat	Kim	Bio
Mesin	0,071	0,071	0,071	0,071	0,071	0,071	0,071	0,071	0,071	0,071	0,214
Industri	0,118	0,118	0,059	0,059	0,118	0,118	0,118	0,118	0,118	0,118	0,059
Kimia	0,103	0,103	0,069	0,069	0,103	0,103	0,103	0,103	0,103	0,103	0,069

4.2.2.4 Menghitung Vektor (S_i)

Untuk menghitung (S_i) dengan cara nilai akademik calon mahasiswa yang sudah diberi bobotkan akan menjadi nilai pangkat dari nilai perbaikan bobot kriteria (W) yang sudah dicari sebelumnya, dapat dilihat pada Persamaan 2.8 dan

dapat dicontohkan pada perhitungan vektors (S_i) di jurusan teknik mesin sebagai berikut:

- $S_i =$
 Data Nilai Agama *perbaikan bobot kriteria jurusan mesin pada mata pelajaran Agama* *
 Data Nilai Kwn *perbaikan bobot kriteria jurusan mesin pada mata pelajaran Kwn* * *
 Data Nilai Biologi *perbaikan bobot kriteria jurusan mesin pada mata pelajaran Biologi*
- $S_i = 4,3119$

Hasil perhitungan Vektor (S_i) pada masing-masing jurusan per mata pelajaran secara lengkap dapat dilihat pada Tabel 4.13.

Tabel 0.13 Menghitung Vektor S

Jurusan	Hasil Vektor S
Teknik Mesin	4,3119
Teknik Industri	4,3114
Teknik Kimia	4,2678

4.2.2.5 Menghitung Vektor (V_i)

Untuk menghitung vektor (V_i) dengan cara nilai vektor (S_i) yang sebelumnya sudah di cari pada tiap jurusan dibagi dengan jumlah seluruh nilai vektor (S_i) yang terdapat pada Persamaan 2.9. Untuk contoh menghitung Vektor (S_i) sebagai berikut:

- $V_i = \frac{\text{Nilai Vektor } S}{\text{Total Seluruh nilai Vektor } S}$
- $V_i = \frac{\text{Nilai Vektor } S \text{ dari Teknik Mesin}}{\text{Total Seluruh nilai Vektor } S}$
- $V_i = \frac{4,3369}{12,8648}$
- $V_i = 0,33449$

Hasil perhitungan Vektor (V_i) pada masing-masing jurusan per mata pelajaran secara lengkap dapat dilihat pada Tabel 4.14.

Tabel 0.14 Menghitung Vektor V

Jurusan	Hasil Vektor V
Teknik Mesin	0,33449
Teknik Industri	0,33445
Teknik Kimia	0,33107

Kemudian setelah diketahui vektor V dari tiap jurusan, dilakukan perbandingan untuk mengetahui jurusan dengan persentase total terbesar. Dari Tabel di atas dapat dilihat bahwa jurusan dengan persentase total terbesar adalah

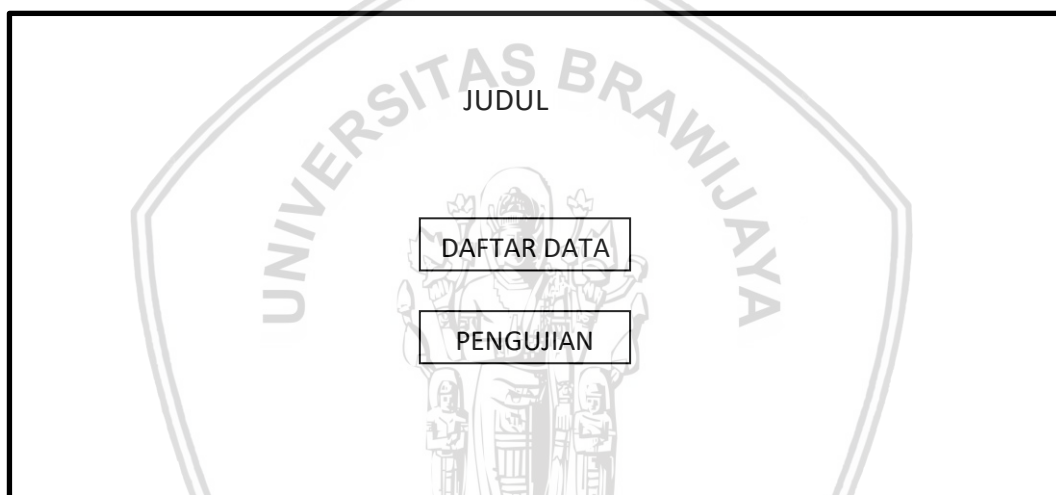
jurusan Mesin , sehingga hasil rekomendasi jurusan untuk calon mahasiswa tersebut adalah Mesin.

4.3 Perancangan Antar Muka

Pada perancangan antarmuka sistem ini memiliki 7 tampilan halaman, yaitu tampilan utama, tampilan daftar data, tampilan pengaturan data uji, tampilan pengujian *Naïve Bayes*, tampilan manualisasi *Naïve Bayes*, tampilan pengujian *Weighted Product*, dan tampilan manualisasi *Weighted Product*.

4.3.1 Perancangan Tampilan Utama

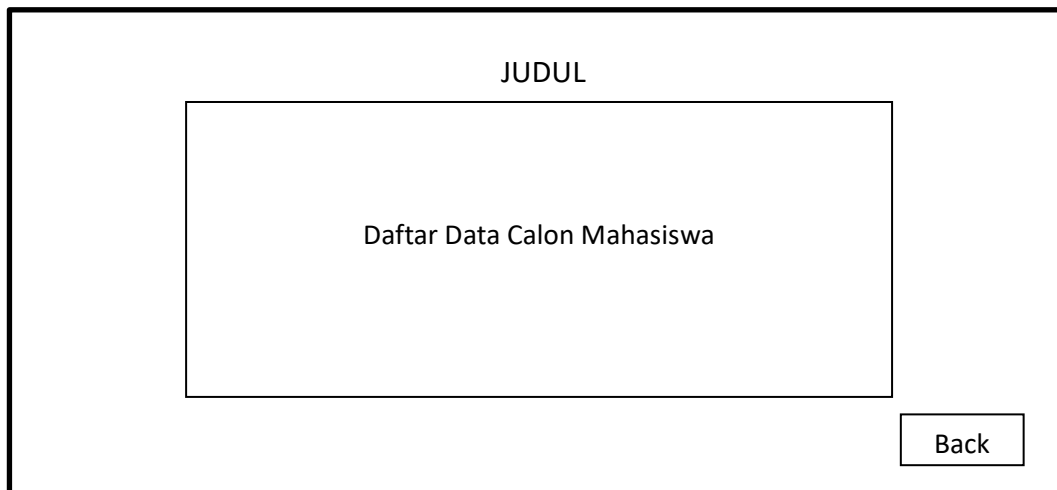
Perancangan tampilan utama akan menampilkan judul dari aplikasi, tombol daftar data untuk menampilkan daftar data yang digunakan untuk perhitungan sistem, tombol pengujian digunakan untuk menampilkan tampilan pengujian *Naïve Bayes*. Perancangan tampilan utama dapat dilihat pada Gambar 4.17.



Gambar 0.17 PerancanganTampilan Utama

4.3.2 Perancangan Tampilan Daftar Data

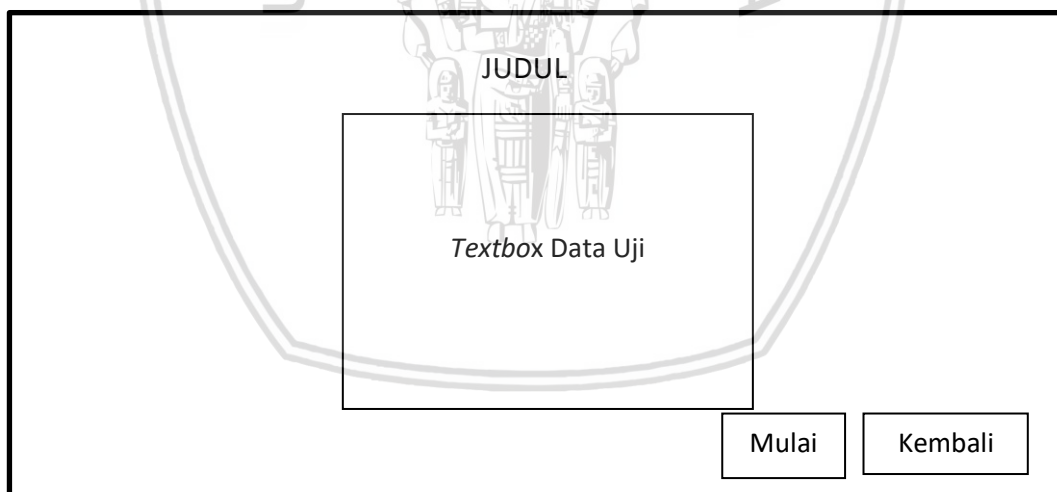
Perancangan tampilan daftar data akan menampilkan judul tampilan dandata yang akan digunakan untuk proses perhitungan pada sistem, tombol kembali untuk kemabli ke tampilan utama. data yang akan ditampilkan adalah data calon mahasiswa yang memiliki atribut nilai akademik dari tiap mata pelajaran beserta keterangan diterima atau ditolak calon mahasiswa tersebut di fakultas tertentu. Peracangan tampilan daftar data dapat dilihat pada Gambar 4.18.



Gambar 0.18 PerancanganTampilan DaftarData

4.3.3 Perancangan Tampilan Pengaturan Data Uji

Perancangan tampilan pengaturan data uji akan menampilkan judul tampilan, *textbox* data uji untuk menginputkan data uji yang akan digunakan pada proses perhitungan sistem, tombol mulai digunakan untuk menuju ke tampilan pengujian Naive Bayes, tombol kembali digunakan untuk kembali ke tampilan utama. perancangan tampilan pengaturan data uji dapat dilihat pada Gambar 4.19.

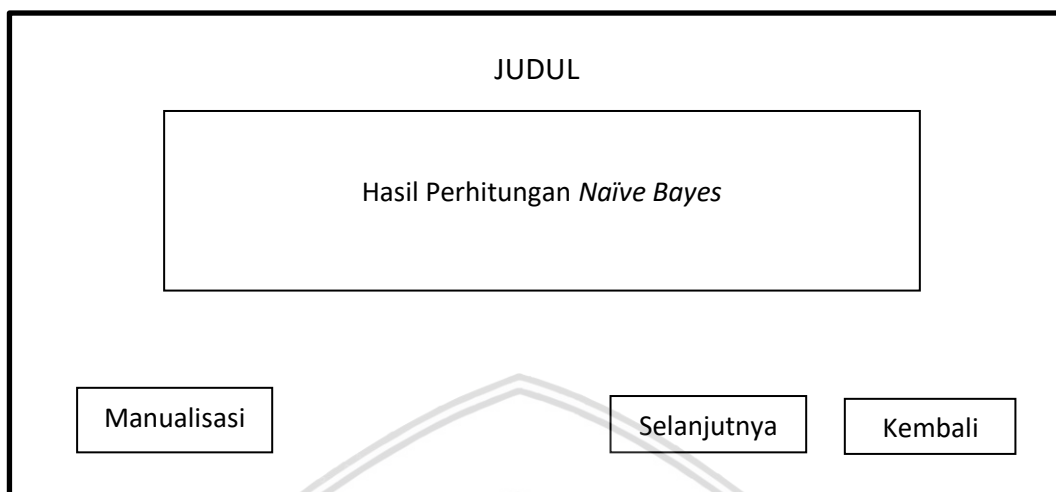


Gambar 0.19 PerancanganTampilan Pengaturan Data Uji

4.3.4 Perancangan Tampilan Pengujian *Naïve Bayes*

Perancangan tampilan pengujian *Naïve Bayes* akan menampilkan judul tampilan, hasil *Naïve Bayes* menampilkan peluang calon mahasiswa diterima di tiap fakultas, tombol selanjutnya untuk menampilkan tampilan WP, tombol

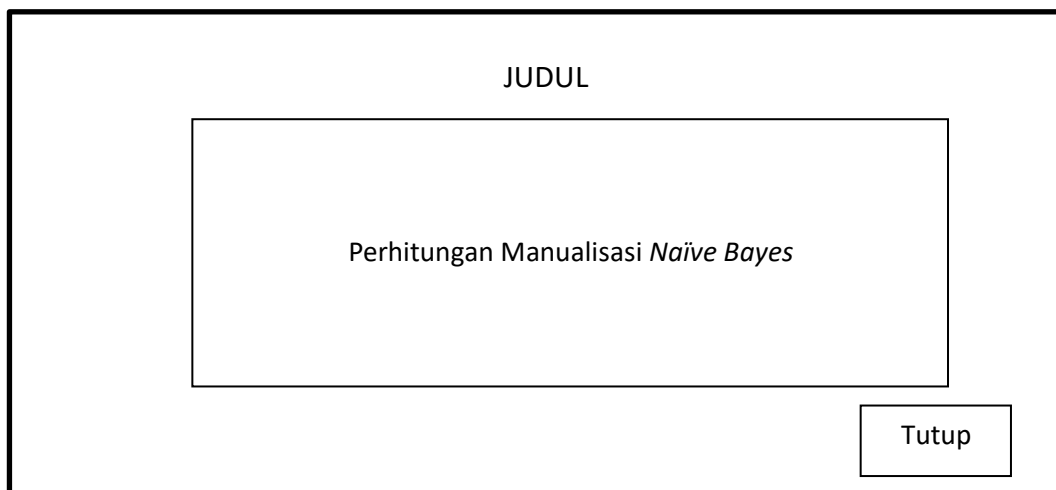
manualisasi untuk menampilkan tampilan manualisasi *Naïve Bayes*, tombol kembali digunakan untuk kembali ke tampilan pengaturan data uji. Perancangan tampilan pengujian *Naïve Bayes* dapat dilihat pada Gambar 4.20.



Gambar 0.20 PerancanganTampilan Pengujian *Naïve Bayes*

4.3.5 Perancangan Tampilan Manualisasi *Naïve Bayes*

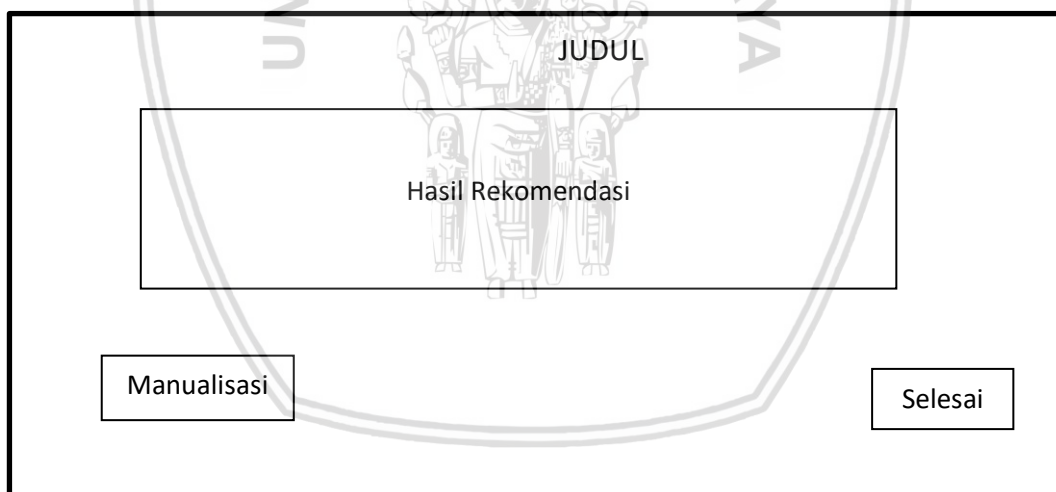
Perancangan tampilan manualisasi *Naïve Bayes* akan menampilkan judul tampilan, perhitungan manualisasi *Naïve Bayes* untuk menampilkan probabilitas diterima calon mahasiswa di tiap fakultas, dan tombol tutup digunakan untuk menutup tampilan manualisasi *Naïve Bayes*. Perancangan tampilan manualisasi *Naïve Bayes* dapat dilihat pada Gambar 4.21.



Gambar 0.21 Perancangan Tampilan Manualisasi *Naïve Bayes*

4.3.6 Perancangan Tampilan Pengujian *Weighted Product* (WP)

Perancangan tampilan pengujian WP menampilkan judul tampilan, hasil rekomendasi WP menampilkan hasil rekomendasi jurusan untuk calon mahasiswa, tombol manualisasi untuk menampilkan tampilan manualisasi WP, tombol selesai digunakan untuk kembali ke tampilan utama. Perancangan tampilan pengujian WP dapat dilihat pada Gambar 4.22.

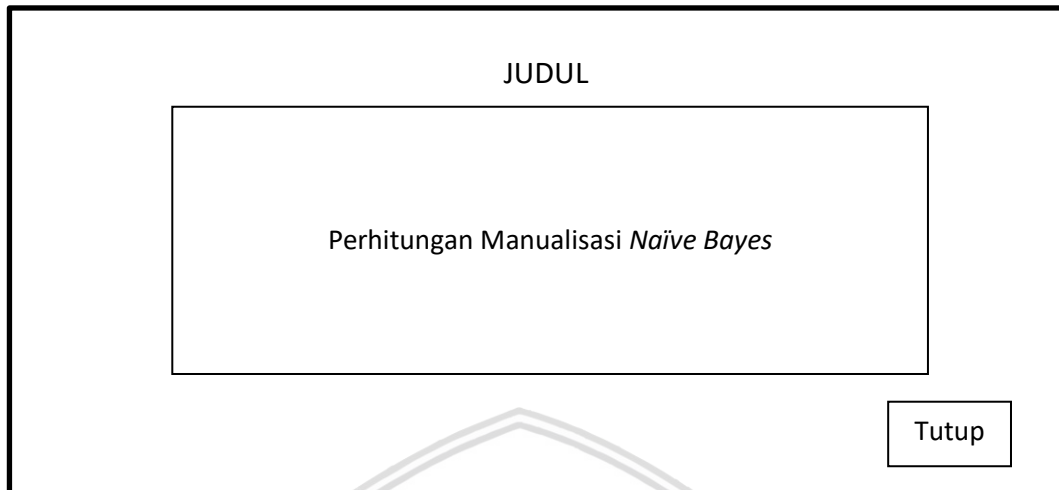


Gambar 0.22 Perancangan Tampilan Pengujian WP

4.3.7 Perancangan Tampilan Manualisasi *Weighted Product* (WP)

Perancangan tampilan manualisasi WP menampilkan judul tampilan, perhitungan manualisasi WP digunakan untuk menentukan hasil rekomendasi jurusan kuliah berdasarkan hasil klasifikasi fakultas menggunakan metode *Naïve*

Bayes, tombol tutup digunakan untuk menutup tampilan manualisasi Weighetd Product. Perancangan tampilan manualisasi WP dapat dilihat pada Gambar 4.23.



Gambar 0.23 PerancanganTampilan Manualisasi WP

Perancangan uji coba dan evaluasi disebabkan karena jumlah data yang sangat berpengaruh pada metode *Naïve Bayes*. Oleh karena itu, untuk melakukan evaluasi sistem maka perlu melakukan uji coba jumlah data latih pada Algoritme *Naïve Bayes*.

4.3.8 Uji Coba Jumlah Data Latih dan Data Uji Algoritme *Naïve Bayes*

Uji coba jumlah data latih dan data uji dengan cara mengubah banyanya jumlah data latih dan data uji pada percobaan, perlunya mengubah banyaknya jumlah data latih dan data uji untuk mendapatkan jumlah data latih dan data uji yang optimal untuk tiap fakultas, untuk mendapatkan hasil klasifikasi yang baik. Rancangan uji coba untuk mengetahui jumlah data latih yang paling optimal dari tiap fakultas dapat dilihat pada Tabel 4.15.

Tabel 0.15 Data Uji Sampel

Jumlah Data Latih (persen)	Jumlah Data Uji (persen)	Percobaan			Rata-Rata Akurasi
		1	2	3	
50	50				
60	40				
70	30				
80	20				

90	10				
100	100				

4.3.9 Uji Coba Jumlah Data Latih dan Data Uji Algoritme *Naïve Bayes* dengan Data Diskrit

Uji coba jumlah data latih dan data uji dengan data diskrit memiliki cara yang sama dengan uji coba jumlah data latih dan data uji algoritme *Naïve Bayes*, perbedaannya adalah dengan mengubah data latih dan data uji ke dalam bentuk diskrit tujuannya untuk membandingkan akurasi sistem yang memiliki data latih dan data uji normal dengan data latih dan data uji diskrit. Rancangan uji coba untuk mengetahui jumlah data latih dan data uji dengan data diskrit yang paling optimal dapat dilihat pada Tabel 4.16.

Tabel 0.16 Data Uji Sampel dengan Mengubah menjadi Data Diskrit

Jumlah Data Latih (persen)	Jumlah Data Uji (persen)	Percobaan			Rata-Rata Akurasi
		1	2	3	
50	50				
60	40				
70	30				
80	20				
90	10				
100	100				

4.3.10 Uji Coba Bobot Nilai Minimum Tiap Jurusan pada Algoritme *Weighted Product (WP)*

Pada pengujian bobot nilai minimum tiap jurusan pada algoritme WP dengan mengubah nilai bobot bertujuan untuk mendapatkan nilai bobot optimal agar mendapatkan hasil klasifikasi dan tingkat akurasi yang terbaik. Di dalam pengujian ini dilakukan dengan menggunakan nilai bobot yang berbeda. Rancangan uji coba bobot nilai minimum tiap jurusan pada algoritme WP dapat dilihat pada Tabel 4.17.

Tabel 0.17 Data Uji Perubahan Bobot

Variasi Bobot Nilai Minimum	Percobaan			Rata-Rata Akurasi
	1	2	3	
Variasi 1				
Variasi 2				
Variasi 3				
Variasi 4				
Variasi 5				



BAB 5 IMPLEMENTASI

Pada bab implementasi ini membahas perancangan sistem pengklasifikasian untuk menentukan pemilihan jurusan kuliah untuk pelajar SMA menggunakan algoritme *Naïve Bayes-WP*. Bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi *Source Code*, dan implementasi antar muka.

5.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan mengacu kepada spesifikasi sistem yang akan digunakan oleh penulis untuk pengimplementasiannya. Spesifikasi sistem dibagi menjadi 2 yaitu, spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pengklasifikasian untuk menentukan pemilihan jurusan kuliah untuk pelajar siswa SMA menggunakan algoritme *Naïve Bayes-WP* ini menggunakan komputer dengan spesifikasi perangkat keras yang digunakan sebagai berikut :

- a. Processor Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz
- b. Kapasitas Memori (RAM) sebesar 4.00 GB

5.1.2 Spesifikasi Perangkat Lunak

Pengklasifikasian untuk menentukan pemilihan jurusan kuliah untuk pelajar siswa SMA menggunakan algoritme *Naïve Bayes-WP* ini menggunakan komputer dengan spesifikasi perangkat lunak yang digunakan sebagai berikut :

- a. Sistem Operasi Windows 10 Enterprise 64 bit
- b. Bahasa Pemrograman Java
- c. *Tools* pemrograman Netbeans 8.1

5.1.3 Batasan-Batasan Implementasi

Batasan implementasi merupakan batasan dari proses yang dilakukan oleh sistem dimana sesuai dengan perancangan yang dibangun. Terdapat beberapa batasan didalam pengimplementasian sistem ini. Berikut merupakan beberapa batasan pengklasifikasian untuk menentukan pemilihan jurusan kuliah untuk pelajar siswa SMA menggunakan algoritme *Naïve Bayes-WP*:

1. Data yang digunakan sebagai data latih dan data uji diperoleh dari data nilai akademik calon mahasiswa. Nilai yang dimasukkan berasal dari nilai rata-rata dari semester 3 hingga semester 5 calon mahasiswa tersebut.

2. *Output* yang dihasilkan adalah hasil rekomendasi fakultas dan jurusan untuk calon mahasiswa.
3. Pengembangan sistem emnggunakan bahasa pemrograman java.
4. Algoritme yang digunakan pada sistem ini adalah *Naïve Bayes* dan *Weighted Product*.

5.2 Implementasi Algoritme

Sistem pengklasifikasian untuk menentukan rekomndasi pemilihan jurusan kuliah untuk pelajar siswa SMA menggunakan algoritme *Naïve Bayes*-WP memiliki 2 implementasi algoritme yaitu, proses *Naïve Bayes* dan Weigthted Product.

5.2.1 Implementasi Algoritme *Naïve Bayes*

Implementasi algoritme *Naïve Bayes* terbagi menjadi beberapa bagian, yaitu:

a. Proses Pendefinisian Data

Di dalam program terdapat *Source Code* untuk mendefinisikan data dengan cara melakukan pembacaan file berekstensi .txt oleh *BufferedReader*. Data dari hasil pembacaan tersebut kemudian dipidahkan berdasarkan tanda ‘ , ‘ dan baris lalu disimpan pada *ArrayList*. Implementasi proses menyimpan data pada *ArrayList* dapat dilihat pada *Source Code* 5.1.

```

1  list = new ArrayList<>();
2  File data = new File(alamatData);
3  BufferedReader bacaData = new BufferedReader(new
4      FileReader(data));
5
6  String baris;
7  while ((baris = bacaData.readLine()) != null) {
8      String[] pisahData = baris.split(",");
9      float[] konversiData = new float[pisahData.length];
10
11     for (int i = 0; i < pisahData.length; i++) {
12         konversiData[i] = Float.valueOf(pisahData[i]);}
13
14     Pelajar p = null;
15     if (konversiData.length == 14) {
16         p = new Pelajar(konversiData[0],
17             konversiData[1], konversiData[2],
18             konversiData[3], konversiData[4],
19             konversiData[5], konversiData[6],
20             konversiData[7], konversiData[8],
21             konversiData[9], konversiData[10],
22             konversiData[11], konversiData[12],
23             konversiData[13]);
24     } else if (konversiData.length == 11) {
25         p = new Pelajar(konversiData[0],
26             konversiData[1], konversiData[2],

```

```

27         konversiData[3], konversiData[4],
28         konversiData[5], konversiData[6],
29         konversiData[7], konversiData[8],
30         konversiData[9], konversiData[10]);
31     }
32     list.add(p);
33 }
34

```

Source Code 0.1 Pendefinisian Data

Penjelasan *Source Code* 5.1 sebagai berikut:

- 1 : Inisialisasi arraylist dengan nama list
 - 2 : Inisialisasi file data latihan
 - 3-4 : Inisialisasi bufferedreader untuk membaca file data latihan
 - 6 : Perulangan untuk membaca file data latihan
 - 7 : Penyimpanan isi file data latihan perbaris ke dalam array pisahData
 - 8 : Inisialisasi array konversiData
 - 11-12 : Perulangan untuk mengisi array konversiData dengan melakukan konversi nilai pisahData dari string ke float pada setiap elemennya
 - 14 : Inisialisasi Pelajar p
 - 15-23 : Kondisi jika panjang array konversiData sama dengan 14 maka p diinstansiasi ke konstruktor dengan 14 parameter
 - 24-30 : Kondisi jika panjang array konversiData sama dengan 11 maka p diinstansiasi ke konstruktor dengan 11 parameter
 - 32 : Penambahan data p ke list
- b. Proses Perhitungan *Naïve Bayes*

Di dalam program terdapat *Source Code* untuk melakukan perhitungan hasil klasifikasi menggunakan *Naïve Bayes*. Proses perhitungan dilakukan dengan mencari mean, standar deviasi, dan likelihood tiap atribut dan tiap kelas. Kemudian membandingkan nilai posterior pada tiap fakultas. Fakultas yang memiliki nilai posterior total tertinggi akan menjadi hasil klasifikasi. Proses perhitungan hasil *Naïve Bayes* dapat dilihat pada *Source Code* 5.2.

```

1 PelajarDAO dao = new PelajarDAO(alamatData);
2 dataLatih = dao.getList();
3 for (Pelajar p : dataLatih) { (
4     if (p.getKategori() == 1 && p.getFakultas() == 2) {
5         jumlahBerhasilTek++;
6     } else if (p.getKategori() == 1 && p.getFakultas() == 3) {
7         jumlahBerhasilKed++;
8     }

```



```

9      }
10     probBerhasilTek = (double) jumlahBerhasilTek / (double)
11         dataLatih.size();
12     probBerhasilKed = (double) jumlahBerhasilKed / (double)
13         dataLatih.size();
14     for (Pelajar p : dataLatih) {
15         if (p.getKategori() == 0 && p.getFakultas() == 2) {
16             jumlahGagalTek++;
17         } else if (p.getKategori() == 0 && p.getFakultas() == 3) {
18             jumlahGagalKed++;
19         }
20     }
21     probGagalTek = (double) jumlahGagalTek / (double)
22     dataLatih.size();
23     probGagalKed = (double) jumlahGagalKed / (double)
24     dataLatih.size();
25     private void meanAgama(double kategori, double fakultas) {
26         meanAgama = 0;
27         meanAgamaT = 0;
28         jumlahAgama = 0;
29         jumlahAgamaTidak = 0;
30         for (Pelajar p : dataLatih) {
31             if (p.getKategori() == 1 && p.getFakultas() == 2)
32             {
33                 meanAgama += p.getnAgm();
34                 jumlahAgama++;
35             } else if (p.getKategori() == 0 && p.getFakultas() == 2) {
36                 meanAgamaT += p.getnAgm();
37                 jumlahAgamaTidak++;
38             } else if (p.getKategori() == 1 && p.getFakultas()
39 == 3) {
40
41                 meanAgama += p.getnAgm();
42                 jumlahAgama++;
43             } else if (p.getKategori() == 0 && p.getFakultas()
44 == 3) {
45                 meanAgamaT += p.getnAgm();
46                 jumlahAgamaTidak++;}}
47         meanAgama /= jumlahAgama;
48         meanAgamaT /= jumlahAgamaTidak;
49     }
50     private void devAgama(double kategori, double fakultas) {
51         devAgama = 0;
52         devAgamaT = 0;
53         meanAgama(kategori, fakultas);
54         for (Pelajar p : dataLatih) {
55             if (p.getKategori() == 1 && p.getFakultas()
56 == 2) {
57                 f
58                 devAgama += (double)
59                 (Math.pow((p.getnAgm() - meanAgama), 2));
60             } else if (p.getKategori() == 0 &&
61 p.getFakultas() == 2) {
62                 devAgamaT += (double) (Math.pow((p.getnAgm()
63 - meanAgamaT), 2));

```

```

63         } else if (p.getKategori() == 1 && p.getFakultas()
64 == 3) {
65             devAgama += (double) (Math.pow((p.getnAgm() -
66 meanAgama), 2));
67         } else if (p.getKategori() == 0 && p.getFakultas()
68 == 3) {
69             devAgamaT += (double) (Math.pow((p.getnAgm()
70 - meanAgamaT), 2));
71         }
72     }
73     devAgama = Math.sqrt(devAgama / (jumlahAgama - 1));
74     devAgamaT = Math.sqrt(devAgamaT / (jumlahAgamaTidak -
75 1));
76 }
77 private double hitungLikelihoodAgamaTek(double kategori,
78 double nilai, double fakultas) {
79
80     devAgama(kategori, fakultas);
81
82     if (kategori == 1 && fakultas == 2) {
83         lhAgm = 0;
84         lhAgm += (double) (1 / (Math.sqrt(2 * Math.PI *
85 Math.pow(devAgama, 2)))
86 * Math.pow(Math.E, -(Math.pow(nilai -
87 meanAgama, 2) / (2 * Math.pow(devAgama, 2)))));
88     } else if (kategori == 0 && fakultas == 2) {
89         lhAgmT += (double) (1 / (Math.sqrt(2 * Math.PI *
90 Math.pow(devAgamaT, 2)))
91 * Math.pow(Math.E, -(Math.pow(nilai -
92 meanAgamaT, 2) / (2 * Math.pow(devAgamaT, 2)))));
93     } else if (kategori == 1 && fakultas == 3) {
94         lhAgm = 0;
95         lhAgm += (double) (1 / (Math.sqrt(2 * Math.PI *
96 Math.pow(devAgama, 2)))
97 * Math.pow(Math.E, -(Math.pow(nilai -
98 meanAgama, 2) / (2 * Math.pow(devAgama, 2)))));
99     }
100     if (kategori == 1) {
101         return (double) (lhAgm);
102     } else {
103         return (double) (lhAgmT);
104     }
105 }
106
107 for (Pelajar p : dataUji) {
108     posteriorTeknik = hitungLikelihoodAgamaTek(1,
109 p.getnAgm(), 2) * hitungLikelihoodKwnTek(1, p.getnKwn(), 2) *
110 hitungLikelihoodIndTek(1, p.getnInd(), 2)
111 * hitungLikelihoodIngTek(1, p.getnIng(),
112 2) * hitungLikelihoodSjrTek(1, p.getnSjr(), 2) *

```

```

116 hitungLikelihoodSnbTek(1, p.getnSnb(), 2)
117         * hitungLikelihoodPjoTek(1, p.getnPjo(),
118 2) * hitungLikelihoodFisTek(1, p.getnFis(), 2) *
119 hitungLikelihoodMatTek(1, p.getnMat(), 2)
120         * hitungLikelihoodKimTek(1, p.getnKim(),
121 2) * hitungLikelihoodBioTek(1, p.getnBio(), 2) *
122 probBerhasilTek;
123     posteriorKedokteran = hitungLikelihoodAgamaTek(1,
124 p.getnAgm(), 3) * hitungLikelihoodKwnTek(1, p.getnKwn(), 3) *
125 hitungLikelihoodIndTek(1, p.getnInd(), 3) *
126 hitungLikelihoodIngTek(1, p.getnIng(), 3) *
127 hitungLikelihoodSjrTek(1, p.getnSjr(), 3) *
128 hitungLikelihoodSnbTek(1, p.getnSnb(), 3) *
129 hitungLikelihoodPjoTek(1, p.getnPjo(), 3) *
130 hitungLikelihoodFisTek(1, p.getnFis(), 3) *
131 hitungLikelihoodMatTek(1, p.getnMat(), 3
132         * hitungLikelihoodKimTek(1, p.getnKim(),
133 3) * hitungLikelihoodBioTek(1, p.getnBio(), 3) *
134 probBerhasilKed;
135     posteriorGagalTek = hitungLikelihoodAgamaTek(0,
136 p.getnAgm(), 2) *
137 hitungLikelihoodKwnTek(0, p.getnKwn(), 2) *
138 hitungLikelihoodIndTek(0, p.getnInd(), 2)
139         * hitungLikelihoodIngTek(0, p.getnIng(),
140 2) * hitungLikelihoodSjrTek(0, p.getnSjr(), 2) *
141 hitungLikelihoodSnbTek(0, p.getnSnb(), 2)
142         * hitungLikelihoodPjoTek(0, p.getnPjo(),
143 2) * hitungLikelihoodFisTek(0, p.getnFis(), 2) *
144 hitungLikelihoodMatTek(0, p.getnMat(), 2)
145         * hitungLikelihoodKimTek(0, p.getnKim(),
146 2) * hitungLikelihoodBioTek(0, p.getnBio(), 2) * probGagalTek;
147     posteriorGagalKed = hitungLikelihoodAgamaTek(0,
148 p.getnAgm(), 3) * hitungLikelihoodKwnTek(0, p.getnKwn(), 3) *
149 hitungLikelihoodIndTek(0, p.getnInd(), 3)
150         * hitungLikelihoodIngTek(0, p.getnIng(),
151 3) * hitungLikelihoodSjrTek(0, p.getnSjr(), 3) *
152 hitungLikelihoodSnbTek(0, p.getnSnb(), 3)
153         * hitungLikelihoodPjoTek(0, p.getnPjo(),
154 3) * hitungLikelihoodFisTek(0, p.getnFis(), 3) *
155 hitungLikelihoodMatTek(0, p.getnMat(), 3) *
156 hitungLikelihoodKimTek(0, p.getnKim(), 3) *
157 hitungLikelihoodBioTek(0, p.getnBio(), 3) * probGagalKed;
158     double[] probabilitas = {posteriorTeknik,
159 posteriorKedokteran, posteriorGagalTek, posteriorGagalKed};
160     String[] kats = {"terima", "terima", "gagal",
161 "gagal"};
162     String[] faks = {"Teknik", "FK", "", ""};
163     double max = -99999;
164     String kat = null;
165

```

```

169         String fak = null;
170         String hkat = "gagal";
171         String hfak = "";
172         if (probabilitas[i] > max) {
173             max = probabilitas[i];
174             kat = kats[i];
175             fak = faks[i];}}
176     public static void main(String[]args) throws Exception {
177         Klasifikasi k = new Klasifikasi();
178         k.latih("C:\\Users\\Asus\\Documents\\NetBeansProjects\\Skrip
179         isManualisasi\\src\\skripismanualisasi\\dataLatih.txt");
180         k.uji("C:\\Users\\Asus\\Documents\\NetBeansProjects\\Skripis
181         Manualisasi\\src\\skripismanualisasi\\dataUji.txt");}
182
183

```

Source Code 0.2 Perhitungan Naïve Bayes

Penjelasan Source Code 5.2 sebagai berikut:

- 1 :Instansiasi objek PelajarDaO
- 2 :Inisialisai variabel data latih
- 3 :Perulangan untuk setiap objek (pelajar)
- 4-5 :Kondisi jika kategori pelajar diterima dan fakultas pelajar teknik maka variabel jumlahBerhasilTek dilakukan increment
- 6-7 :Kondisi jika kategori pelajar diterima dan fakultas pelajar kedokteran maka variabel jumlahBerhasilKed dilakukan increment
- 10-11 :Perhitungan untuk probabilitas berhasil teknik
- 12-13 :Perhitungan untuk probabilitas berhasil kedokteran
- 14 :Perulangan untuk setiap objek (pelajar)
- 15-16 :Kondisi jika kategori pelajar tidak diterima dan fakultas pelajar teknik maka variabel jumlahGagalTek dilakukan increment
- 17-18 :Kondisi jika kategori pelajar tidak diterima dan fakultas pelajar kedokteran maka variabel jumlahGagalKed dilakukan increment
- 21-22 :Perhitungan untuk probabilitas gagal teknik
- 23-24 :Perhitungan untuk probabilitas gagal kedokteran
- 25 :Deklarasi method meanAgama untuk menghitung mean
- 26-29 :Inisialisasi variabel meanAgama, meanAgamaT, jumlahAgama, jumlahAgamaT dengan 0
- 30 :Perulangan untuk setiap objek (pelajar) sebanyak data latih
- 31-34 :Kondisi jika kategori pelajar diterima fakultas teknik maka dilakukan penambahan untuk mean nilai agama dan increment variabel jumlahAgama
:Kondisi jika kategori pelajar tidak diterima fakultas teknik maka dilakukan penambahan untuk mean nilai agamaT dan increment variabel jumlahAgamaTidak
- 35-37

38-42	:Kondisi jika kategori pelajar diterima fakultas kedokteran maka dilakukan penambahann untuk mean nilai agama dan increment variabel jumlahAgama
43-46	:Kondisi jika kategori tidak pelajar diterima fakultas kedokteran maka dilakukan penambahan untuk mean nilai agamaT dan increment variabel jumlahAgamaTidak
	:Perhitungan untuk mean nilai agama
47	:Perhitungan untuk mean nilai agamaT
48	:Deklarasi method devAgama untuk menghitung standar deviasi
50	:Inisialisasi variabel devAgama dan devAgamaT dengan 0
51-52	:Pemanggilan method meanAgama
53	:Perulangan untuk setiap objek (pelajar) sebanyak data latih
54	:Kondisi jika kategori pelajar diterima fakultas teknik maka dilakukan
55-58	penambahan untuk standar deviasi nilai agama
	:Kondisi jika kategori pelajar tidak diterima fakultas teknik maka
59-62	dilakukan penambahan untuk standar deviasi nilai agamaT
	:Kondisi jika kategori pelajar diterima fakultas kedokteran maka
63-66	dilakukan penambahann untuk standar deviasi nilai agama
	:Kondisi jika kategori tidak pelajar diterima fakultas kedokteran maka
67-70	dilakukan penambahan untuk standar deviasi nilai agamaT
	:Perhitungan untuk standar deviasi nilai agama
73	:Perhitungan untuk standar deviasi nilai agamaT
74-75	:Deklarasi method hitungLilihoodAgamaTek
77-79	:Pemanggilan method devAgama
80-81	:Kondisi jika kategori pelajar diterima fakultas teknik maka dilakukan
82-87	penambahan untuk likelihood nilai agama
88-93	:Kondisi jika kategori pelajar tidak diterima fakultas teknik maka
	dilakukan penambahan untuk likelihood nilai agamaT
94-102	:Kondisi jika kategori pelajar diterima fakultas kedokteran maka
	dilakukan penambahann untuk likelihood nilai agama Kondisi jika
	kategori tidak pelajar diterima fakultas kedokteran maka dilakukan
	penambahan untuk likelihood nilai agamaT
103-105	:Kondisi jika kategori sama dengan 1 maka dilakukan pengembalian
	nilai dari likelihood nilai agama
106-109	:Kondisi jika kategori selain sama dengan 1 maka dilakukan
	pengembalian nilai dari likelihood nilai agamaT
110	:Perulangan untuk setiap objek (pelajar) sebanyak data uji
111-122	:Perhitungan untuk posterior berhasil teknik
123-136	:Perhitungan untuk posterior berhasil kedokteran
137-148	:Perhitungan untuk posterior gagal tekni
149-160	:Perhitungan untuk posterior gagal kedokteran
161-162	:Inisialisasi array probabilitas

- 163-164 :Inisialisasi array kats untuk daftar kategori
- 165 :Inisialisasi array faks untuk daftar fakultas
- 166-172 :Inisialisasi variabel max, kat, fak, hkat, hfak
- 173-176 :Perulangan untuk mencari nilai probabilitas terbesar
- 177 :Deklarasi method main
- 178 :Instansiasi objek k dari kelas Klasifikasi
- 179-180 :Pemanggilan method latih oleh k
- 181-183 :Pemanggilan method uji oleh k

5.2.2 Implementasi Algoritme *Weighted Product* (WP)

Impelentasi algoritme WP terbagi menjadi beberapa bagian, yaitu:

c. Proses pengambilan data nilai calon mahasiswa

Di dalam program tersebut terdapat *Source Code* untuk mengambil data dari hasil seleksi menggunakan algoritme *Naïve Bayes* pada file dataLatih.txt . Implementasi dapat dilihat pada *Source Code* 5.3

```

1  ArrayList<Pelajar> dataLatih;
2  ArrayList<double[]> dataBaru = new ArrayList<double[]>();
3  double[][] data;
4  int idFak;
5  public wp(double i) throws Exception {
6      PelajarDAO dao = new PelajarDAO
7          ("C:\\Users\\Asus\\Documents\\NetBeansProjects\\
8          pengujian\\src\\pengujian\\dataLatih.txt");
9      dataLatih = dao.getList();
10     for (Pelajar p : dataLatih) {
11         if (p.getKategori()== 1 && p.getFakultas()== (i+2)){
12             double[] k = new double[14];
13             k[0] = (double) p.getnAgm();
14             k[1] = (double) p.getnKwn();
15             k[2] = (double) p.getnInd();
16             k[3] = (double) p.getnIng();
17             k[4] = (double) p.getnSjr();
18             k[5] = (double) p.getnSnb();
19             k[6] = (double) p.getnPjo();
20             k[7] = (double) p.getnFis();
21             k[8] = (double) p.getnMat();
22             k[9] = (double) p.getnKim();
23             k[10] = (double) p.getnBio();
24             k[11] = (double) p.getKategori();
25             k[12] = (double) p.getFakultas();
26             k[13] = (double) p.getJurusan();
27             dataBaru.add(k);
28         }
29     }
30     data = new double[dataBaru.size()][14];
31     for (int l = 0; l < dataBaru.size(); l++) {
32         data[l] = dataBaru.get(l);
33     }

```


34 }

Source Code 0.3 Pengambilan Data Nilai Calon Mahasiswa

Penjelasan dari *Source Code* 5.3 adalah sebagai berikut:

- 1 : Deklarasi arraylist dengan nama dataLatih
- 2 : Inisialisasi arraylist dengan nama dataBaru
- 3 : Deklarasi array dengan nama data
- 4 : Deklarasi variabel integer dengan nama idFak
- 5 : Deklarasi konstruktor kelas wp
- 6-8 : Instansiasi objek dao dari kelas pelajarDAO
- 9 : Inisialisasi arraylist dataLatih
- 10 : Perulangan objek p sebanyak data latih
- 11-29 : Kondisi jika kategori diterima dan fakultas teknik atau kedokteran maka isi array k dengan nilai tiap elemen pada p dan tambahkan k pada arraylist dataBaru
- 30 : Inisialisasi array data
- 31-34 : Perulangan untuk mengisi array data dengan nilai dari dataBaru tiap barisnya

d. Proses pengambilan data nilai minimum untuk setiap mata pelajaran

Di dalam program ini terdapat *Source Code* untuk mendapatkan nilai minimum untuk setiap mata pelajaran untuk tiap fakultas. Data nilai minimum didapatkan dari file txt. Implementasi dapat dilihat pada *Source Code* 5.4.

```

1 public double[][] minimum() {
2     int n = data.length;
3     int m = data[0].length - 4;
4     double[][] min = new double[3][m];
5     idFak = (int) data[0][12];
6     ArrayList<double[]> indeks0 = new ArrayList<double[]>();
7     ArrayList<double[]> indeks1 = new ArrayList<double[]>();
8     ArrayList<double[]> indeks2 = new ArrayList<double[]>();
9
10    for (int j = 0; (j < n); j++) {
11        switch ((int) data[j][13]) {
12            case (4):
13                indeks0.add(data[j]);
14                break;
15            case (5):
16                indeks1.add(data[j]);
17                break;
18            case (6):
19                indeks2.add(data[j]);
20                break;
21            case (7):
22                indeks0.add(data[j]);
23                break;
24            case (8):

```



```

25         indeks1.add(data[j]);
26         break;
27     case (9):
28         indeks2.add(data[j]);
29         break;
30     }
31 }
32
33 double toArrayindeks0[][] = new
34     double[indeks0.size()][11];
35 double toArrayindeks1[][] = new
36     double[indeks1.size()][11];
37 double toArrayindeks2[][] = new
38     double[indeks2.size()][11];
39 for (int i = 0; i < indeks0.size(); i++) {
40     toArrayindeks0[i] = indeks0.get(i);
41 }
42 for (int i = 0; i < indeks1.size(); i++) {
43     toArrayindeks1[i] = indeks1.get(i);
44 }
45 for (int i = 0; i < indeks2.size(); i++) {
46     toArrayindeks2[i] = indeks2.get(i);
47 }
48
49 double[] minIndeks0 = {999, 999, 999, 999, 999, 999, 999, 999,
50     999, 999, 999};
51 double[] minIndeks1 = {999, 999, 999, 999, 999, 999, 999, 999,
52     999, 999, 999};
53 double[] minIndeks2 = {999, 999, 999, 999, 999, 999, 999, 999,
54     999, 999, 999};
55 for (int i = 0; i < 11; i++) {
56     for (int j = 0; j < toArrayindeks0.length; j++) {
57         if (toArrayindeks0[j][i] < minIndeks0[i]) {
58             minIndeks0[i] = toArrayindeks0[j][i];
59         }
60     }
61 }
62 for (int i = 0; i < 11; i++) {
63     for (int j = 0; j < toArrayindeks2.length; j++) {
64         if (toArrayindeks2[j][i] < minIndeks2[i]) {
65             minIndeks2[i] = toArrayindeks2[j][i];
66         }
67     }
68 }
69 for (int i = 0; i < 11; i++) {
70     for (int j = 0; j < toArrayindeks1.length; j++) {
71         if (toArrayindeks1[j][i] < minIndeks1[i]) {
72             minIndeks1[i] = toArrayindeks1[j][i];
73         }
74     }
75 }
76 min[0] = minIndeks0;
77 min[1] = minIndeks1;
78 min[2] = minIndeks2;

```

79	return (min);
80	}

Source Code 0.4 Pengambilan Data Nilai Minimum

Penjelasan *Source Code* 5.4 adalah sebagai berikut:

- 1 :Deklarasi method minimum
- 2 :Inisialisasi jumlah baris dan kolom dalam variabel n dan m
- 3 :Inisialisasi array min
- 4 :Inisialisasi variabel idFak
- 5 :Inisialisasi arraylist dengan nama indeks0
- 6 :Inisialisasi arraylist dengan nama indeks1
- 8 :Inisialisasi arraylist dengan nama indeks2
- 10 :Perulangan untuk mengisi indeks0, indeks1, dan indeks2
- 11-30 :Switch case jika data baris ke-j kolom ke 13 sama dengan nilai case maka dilakukan penambahan data ke-j ke arraylist indeks0, indeks1, atau indeks2
- 33 :Inisialisasi array dengan nama toArrayIndeks0
- 35 :Inisialisasi array dengan nama toArrayIndeks1
- 37 :Inisialisasi array dengan nama toArrayIndeks2
- 39 :Perulangan untuk mengisi array toArrayIndeks0 dengan nilai tiap baris pada arraylist indeks0
- 42 :Perulangan untuk mengisi array toArrayIndeks1 dengan nilai tiap baris pada arraylist indeks1
- 45 :Perulangan untuk mengisi array toArrayIndeks2 dengan nilai tiap baris pada arraylist indeks2
- 49 :Inisialisasi array minIndeks0
- 51 :Inisialisasi array minIndeks1
- 53 :Inisialisasi array minIndeks2
- 55 :Perulangan untuk mengisi array minIndeks0 dengan mencari nilai terkecil pada setiap kolom di array toIndeks0
- 62 :Perulangan untuk mengisi array minIndeks1 dengan mencari nilai terkecil pada setiap kolom di array toIndeks1
- 69 :Perulangan untuk mengisi array minIndeks2 dengan mencari nilai terkecil pada setiap kolom di array toIndeks2
- 76 :Inisialisai array min indeks ke 0 dengan minIndeks0
- 77 :Inisialisai array min indeks ke 1 dengan minIndeks1
- 78 :Inisialisai array min indeks ke 2 dengan minIndeks2
- 79-80 :Pengembalian nilai min

e. Pembobotan Mata Pelajaran untuk Tiap Fakultas

Di dalam program ini terdapat *Source Code* untuk memberikan bobot pada tiap mata pelajaran pada tiap fakultas. Pemberian bobot dilakukan dengan membandingkan nilai minimum pada setiap mata pelajaran dengan nilai calon mahasiswa pada mata pelajaran yang sama. Implementasi dapat dilihat pada *Source Code 5.5*.

```

1 public int[][] konversiBobot(double data[][], double[] uji)
2 {
3     int n = data.length;
4     int m = data[0].length;
5     int[][] konv = new int[n][m];
6
7     for (int j = 0; j < m; j++) {
8         int nilaidiBawah = 0;
9         double[] selisih = new double[3];
10        for (int i = 0; i < n; i++) {
11            if (uji[j] < data[i][j]) {
12                konv[i][j] = 1;
13                nilaidiBawah++;
14            } else {
15                konv[i][j] = 0;
16            }
17            selisih[i] = uji[j] - data[i][j];
18        }
19        if (nilaidiBawah == 2) {
20            for (int i = 0; i < n; i++) {
21                if (konv[i][j] == 0) {
22                    konv[i][j] = 2;
23                }
24            }
25        } else if (nilaidiBawah <= 1) {
26            Map<Integer, Double> rank = new HashMap<>();
27            int[] rankVal = new int[3];
28            for (int i = 0; i < selisih.length; i++) {
29                if (konv[i][j] == 1) {
30                    selisih[i] = 999;
31                }
32                rank.put(i, selisih[i]);
33            }
34            rankVal = rank.entrySet().stream().
35                sorted(Entry.comparingByValue())
36                .mapToInt(Entry::getKey).toArray();
37
38            int k = 0; // perubahan rank
39            while (k < 3) {
40                if (rankVal[k] + 1 == 3) { //2
41                    konv[k][j] = 1;
42                } else if (rankVal[k] + 1 == 1) { //0
43                    konv[k][j] = 3;
44                } else {
45                    konv[k][j] = rankVal[k] + 1; //1
46                }
47                k++;
48            }
49        }
50    }
51 }

```

46	}
47	}
48	}
49	return (konv);
50	}

Source Code 0.5 Pembobotan Mata Pelajaran untuk Tiap Fakultas

Penjelasan *Source Code* 5.5 adalah sebagai berikut:

- 1 :Deklarasi method konversiBobot
- 3-4 :Inisialisasi jumlah baris dan kolom dalam variabel n dan m
- 5 :Inisialisasi array konv
- 7 :Perulangan untuk mengisi array konv
- 8 :Inisialisasi variabel nilaidiBawah dengan 0
- 9 :Inisialisasi array selisih
- 11 :Kondisi jika data uji lebih kecil dari data latih maka nilai konv dalah 0 dan variabel nilaidiBawah dilakukan increment
- 14 :Kondisi jika data uji lebih besar atau sama dengan dari data latih maka nilai konv dalah 1 dan dilakukan perhitungan selisih antara data uji dan data latih
- 18 :Kondisi jika nilai variabel nilaidiBawah sama dengan 2 maka untuk setiap data konv dengan nilai 0 diganti dengan 2
- 24 :Kondisi jika nilai variabel nilaidiBawah kurang dari sama dengan 1
- 25 :Maka dilakukan perankingan dengan menggunakan hashmap berdasarkan nilai selisih
- 37 :Perulangan untuk membalik hasil perangkingan hashmap dari besar ke kecil menjadi kecil ke besar
- 49-50 :Pengembalian nilai konv

f. Proses Perhitungan *Weighted Product* (WP)

Di dalam program ini terdapat *Source Code* untuk melakukan perhitungan untuk mencari rekomendasi jurusan berdasarkan nilai akademik menggunakan WP dengan cara mencari perbaikan bobot kemudian mencari hasil vektor s dan yang terakhir mencari nilai vektor v. Implementasi dapat dilihat pada *Source Code* 5.6.

1	public double[][] perbaikanBobot(int data[][]) {
2	int n = data.length;
3	int m = data[0].length;
4	double[] sum = new double[n];
5	double[][] w = new double[n][m];
6	for (int i = 0; i < n; i++) {
7	for (int j = 0; j < m; j++) {
8	sum[i] += data[i][j];
9	}
10	}

```

11
12         for (int i = 0; i < n; i++) {
13             for (int j = 0; j < m; j++) {
14                 w[i][j] = data[i][j] / sum[i];
15             }
16         }
17
18         return (w);
19     }
20
21     public double[] pembentukanVektorS(double w[][], double
22     konversi[]) {
23         int n = w.length;
24         int m = w[0].length;
25         double[] vektorS = new double[n];
26         for (int i = 0; i < n; i++) {
27             for (int j = 0; j < m; j++) {
28                 if (j == 0) {
29                     vektorS[i] = Math.pow(konversi[j],
30 w[i][j]);
31                 } else {
32                     vektorS[i] *= Math.pow(konversi[j],
33 w[i][j]);
34                 }
35             }
36         }
37
38         return (vektorS);
39     }
40
41     public double[] pembentukanVektorV(double vektorS[]) {
42         int n = vektorS.length;
43         double[] vektorV = new double[n];
44         double sum = 0;
45         for (int i = 0; i < n; i++) {
46             sum += vektorS[i];
47         }
48
49         for (int i = 0; i < n; i++) {
50             vektorV[i] = vektorS[i] / sum;
51         }
52         return (vektorV);
53     }
54     public double[] pembentukanVektorV(double vektorS[]) {
55         int n = vektorS.length;
56         double[] vektorV = new double[n];
57         double sum = 0;
58         for (int i = 0; i < n; i++) {
59             sum += vektorS[i];
60         }
61
62         for (int i = 0; i < n; i++) {
63             vektorV[i] = vektorS[i] / sum;
64         }

```

65	return (vektorV);
66	}

Source Code 0.6 Perhitungan WP

Penjelasan *Source Code* 5.6 adalah sebagai berikut:

- 1 :Deklarasi method perbaikanBobot
- 2 :Inisialisasi jumlah baris dan kolom dalam variabel n dan m
- 4 :Inisialisasi array sum
- 5 :Inisialisasi array w
- 6 :Perulangan untuk mengisi array sum dengan menambahkan data setiap barisnya
- 12 :Perulangan untuk mengisi array w dengan membagi data dengan nilai sum setiap barisnya
- 18 :Mengembalikan nilai w
- 21 :Deklarasi method pembentukanVektorS
- 23 :Inisialisasi jumlah baris dan kolom dalam variabel n dan m
- 25 :Inisialisasi array vektorS
- 26 :Perulangan untuk mengisi array vektorS dengan kondisi jika nilai j sama dengan 0 maka dilakukan perhitungan vektor S awal dan jika nilai j lebih dari 0 maka perhitungan vektor S dikalikan dengan hasil perhitungan vektorS sebelumnya
- 38 :Pengembalian nilai vektorS
- 41 :Deklarasi method pembentukanVektorV
- 42 :Inisialisasi jumlah baris dalam variabel n
- 43 :Inisialisasi array vektorV
- 44 :Inisialisasi variabel sum dengan 0
- 45 :Perulangan untuk mendapatkan nilai sum dari penjumlahan nilai vektorS pada setiap barisnya
- 49 :Perulangan untuk mengisi array vektorV dengan pembagian nilai vektorS pada setiap barisnya dengan nilai sum
- 52 :Pengembalian nilai vektorS

g. Penentuan Hasil Rekomendasi

Di dalam program ini terdapat *Source Code* untuk menentukan hasil rekomendasi berdasarkan nilai vektor V. Nilai vektor v yang tertinggi maka akan menjadi jurusan rekomendasi utama. Implementasi dapat dilihat pada *Source Code* 5.7.

1	public String penentuanHasil(double data[]) {
2	double[][] minData = minimum();
3	int[][] konvData = konversiBobot(minData, data);
4	double[][] w = perbaikanBobot(konvData);
5	double[] vektorS = pembentukanVektorS(w,
6	konversi(data));

```
7      int n = vektorV.length;
8      String[] hasil = new String[3];
9
10     int[] rankVal = new int[3];
11     for (int i = 0; i < n; i++) {
12         int count = 0;
13         for (int j = 0; j < n; j++) {
14             if (vektorV[j] > vektorV[i]) {
15                 count++;
16             }
17         }
18         rankVal[i] = count + 1;
19     }
20     for (int i = 0; i < n; i++) {
21         switch (rankVal[i]) {
22             case 1:
23                 if (idFak == 2) {
24                     if (i == 0) {
25                         hasil[0] = "mesin";
26                     } else if (i == 1) {
27                         hasil[0] = "industri";
28                     } else if (i == 2) {
29                         hasil[0] = "kimia";
30                     }
31                 } else {
32                     if (i == 0) {
33                         hasil[0] = "dokter";
34                     } else if (i == 1) {
35                         hasil[0] = "farmasi";
36                     } else if (i == 2) {
37                         hasil[0] = "gizi";
38                     }
39                 }
40                 break;
41             case 2:
42                 if (idFak == 2) {
43                     if (i == 0) {
44                         hasil[1] = "mesin";
45                     } else if (i == 1) {
46                         hasil[1] = "industri";
47                     } else if (i == 2) {
48                         hasil[1] = "kimia";
49                     }
50                 } else {
51                     if (i == 0) {
52                         hasil[1] = "dokter";
53                     } else if (i == 1) {
54                         hasil[1] = "farmasi";
55                     } else if (i == 2) {
56                         hasil[1] = "gizi";
57                     }
58                 }
59                 break;
60             case 3:
```



```

61         if (idFak == 2) {
62             if (i == 0) {
63                 hasil[2] = "mesin";
64             } else if (i == 1) {
65                 hasil[2] = "industri";
66             } else if (i == 2) {
67                 hasil[2] = "kimia";
68             }
69         } else {
70             if (i == 0) {
71                 hasil[2] = "dokter";
72             } else if (i == 1) {
73                 hasil[2] = "farmasi";
74             } else if (i == 2) {
75                 hasil[2] = "gizi";
76             }
77         }
78         break;
79     }
80 }
81
82 return ("1. " + hasil[0] + " 2. " + hasil[1] + " 3. "
83 + hasil[2]);
84 }

```

Source Code 0.7 Penentuan Hasil Rekomendasi

Penjelasan Source Code 5.7 adalah sebagai berikut:

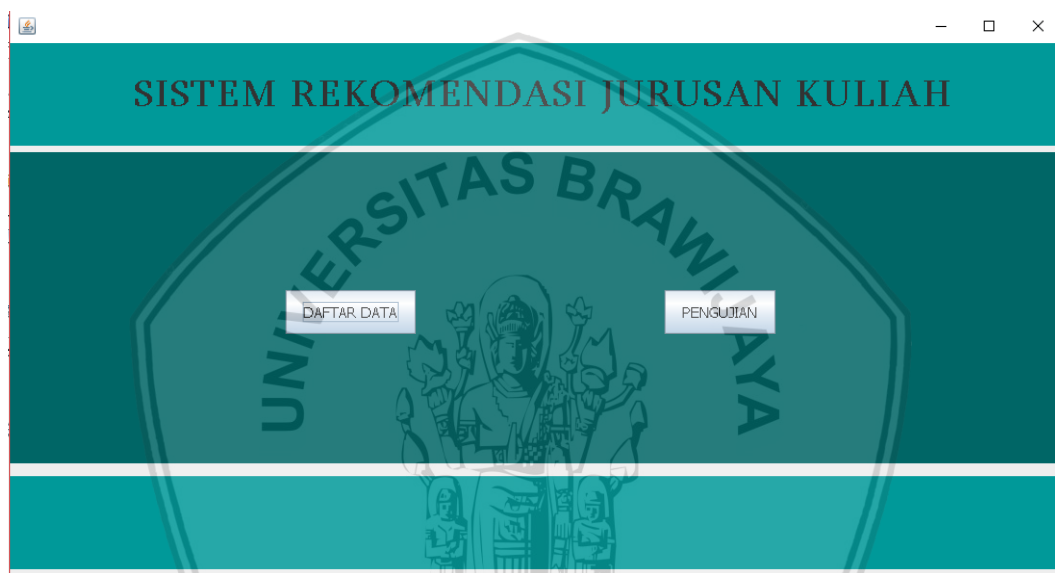
- 1 :Deklarasi method penentuanHasil
- 2 :Inisialisasi array minData dengan pemanggilan method minimum
- 3 :Inisialisasi array konvData dengan pemanggilan method konversiBobot
- 4 :Inisialisasi array w dengan pemanggilan method perbaikanBobot
- 5 :Inisialisasi array vektorS dengan pemanggilan method pembentukan vektorS
- 7 :Inisialisasi jumlah baris dalam variabel n
- 8 :Inisialisasi array hasil
- 10 :Inisialisasi array rankVal
- 11 :Perulangan untuk mengisi array rankVal dengan mencari nilai terbesar pada vektorV
- 20 :Perulangan untuk mengisi array hasil
- 21 :Switch case jika nilai rankVal pada baris i sesuai dengan case 1, 2, atau 3 maka dilakukan pengecekan kondisi jika idFak sama dengan 2 maka hasil pada indeks tersebut adalah mesin, industri, atau kimia. Jika idFak tidak sama dengan 2 maka hasil pada indeks tersebut adalah dokter, farmasi, atau gizi.
- 82 :Pengembalian nilai string berupa urutan hasil

5.3 Implementasi Antar Muka

Pada implementasi antar muka akan dibuat 7 tampilan yaitu tampilan utama, tampilan daftar data, tampilan pengaturan data uji, tampilan pengujian *Naïve Bayes*, tampilan manualisasi *Naïve Bayes*, tampilan pengujian WP, dan tampilan manualisasi WP.

5.3.1 Implementasi Tampilan Utama

Implementasi tampilan utama akan menampilkan judul dari aplikasi, tombol daftar data untuk menampilkan daftar data yang digunakan untuk perhitungan sistem, tombol pengujian digunakan untuk menampilkan tampilan pengujian *Naïve Bayes*. Implementasi tampilan utama dapat dilihat pada Gambar 5.1.



Gambar 0.1 Implementasi Tampilan Utama

5.3.2 Implementasi Tampilan Daftar Data

Implementasi tampilan daftar data akan menampilkan judul tampilan dandata yang akan digunakan untuk proses perhitungan pada sistem, tombol kembali untuk kemabli ke tampilan utama. data yang akan ditampilkan adalah data calon mahasiswa yang memiliki atribut nilai akademik dari tiap mata pelajaran beserta keterangan diterima atau ditolak calon mahasiswa tersebut di fakultas tertentu. Implementasi tampilan daftar data dapat dilihat pada Gambar 5.2.

DAFTAR DATA											
AGM	KWN	BIIND	BING	SJR	SNB	FIS	MAT	KIM	BIO	KATEGORI	FAKULTAS
Data1	87.33	88	89	92.67	91.33	85.67	86	90.67	86.33	90.67	94.67
Data2	89.33	89.67	83.67	90	87.67	88.67	90	92	90.67	92.33	93.67
Data3	91	88.67	81.33	91.67	85.67	95	86	90.33	94	90	90.67
Data4	87	89	80.33	86.67	92	89	86	89	88	90.33	90.33
Data5	87.33	88.67	83	88.67	92	87	96.67	82	86.33	85.67	89.67
Data6	86	87.67	80.33	90.33	85	91	87	89.33	88	90.33	88.67
Data7	89.67	88	80.67	86	83.33	85	88	91	88	89.67	89.67
Data8	95.33	95.33	89	88	93	91.33	96.67	87.67	94	94	92.33
Data9	92.33	91.33	89	92	97.67	89	87.33	94.33	91	96.33	96
Data10	88	90.33	84.33	93.33	89.67	97.67	87.67	94	93.33	96.67	92
Data11	94	93.67	91.33	89	93	90.33	89.33	89.33	93	90	92.67
Data12	92.33	93	89.67	91.33	94	86	86	90.33	91	93	93.67
Data13	91	89.33	89.33	84	90	88	86	82	85	85	89
Data14	93.67	93	83.67	90.67	96.33	89.33	86.33	88.67	87	92.67	92
Jumlah Data										125	
KEMBALI											

Gambar 0.2 Implementasi Tampilan Daftar Data

5.3.3 Implementasi Tampilan Data Uji

Implementasi tampilan pengaturan data uji akan menampilkan judul tampilan, *textbox* data uji untuk menginputkan data uji yang akan digunakan pada proses perhitungan sistem, tombol mulai digunakan untuk menuju ke tampilan pengujian Niave Bayes, tombol kembali digunakan untuk kembali ke tampilan utama. Implementasi tampilan data uji dapat dilihat pada Gambar 5.3.

DATA UJI											
AGM	KWN	BIIND	BING	SJR	SNB	FIS	MAT	KIM	BIO		
Data1	93.67	93	83.67	90.67	96.33	89.33	86.33	88.67	87		
Data2	84.67	87.33	88.33	89	91	86.33	85.67	86	87.33		
Data3	90.67	92	88.67	88.33	94.67	84.33	85.67	89	85.67		
Data4	92.67	90	82.33	90.33	97.33	88.67	86.33	87.33	87.67		
Data5	91.67	90.67	85.33	89.67	95.67	90.67	87	83	84.33		
Data6	90.67	88.33	84.67	88.33	87.67	88.33	89	91	90		
Data7	90	89.33	86.33	84	87	91	89	82.33	91.67		
Data8	90.33	87.67	82.33	90.33	87	91	88	89	90		
Data9	89.57	88.33	83	90.33	86.33	94	86.33	90.33	88.33		
Data10	90	89.67	83	90.67	87.33	85.33	87.67	91	88.33		
Data11	85	88	85.67	80	88	85.67	85.33	88.67	81.67		
Data12	90.33	89.33	82.33	90	86.67	93.33	86.33	89.67	81.67		
Data13	86.67	86.33	81.33	87.67	85	96	86.33	85	82.33		
Data14	86.33	86.33	80.67	88	86.67	87.33	88.67	90.33	86		
Data15	83.33	89.67	87	87.67	80	89	86	81.33	82.33		
Data16	86.67	86.33	81.33	87.67	85	96	86.33	85	82.33		
Jumlah Data										25	
HITUNG KEMBALI											

Gambar 0.3 Implementasi Tampilan Data Uji

5.3.4 Implementasi Tampilan Pengujian *Naïve Bayes*

Implementasi tampilan manualisasi *Naïve Bayes* akan menampilkan judul tampilan, perhitungan manualisasi *Naïve Bayes* untuk menampilkan probabilitas diterima calon mahasiswa di tiap fakultas, dan tombol tutup digunakan untuk

menutup tampilan manualisasi *Naïve Bayes*. Implementasi tampilan hasil pengujian *Naive Bayes* dapat dilihat pada Gambar 5.4.

Data	Hasil
Data1	Teknik
Data2	Teknik
Data3	Teknik
Data4	Teknik
Data5	Teknik
Data6	Teknik
Data7	Teknik
Data8	Teknik
Data9	Teknik
Data10	Teknik
Data11	Tolak
Data12	Tolak
Data13	Tolak
Data14	Tolak

Akurasi 68%

MANUALISASI LANJUT KEMBALI

Gambar 0.4 Implementasi Tampilan Pengujian *Naïve Bayes*

5.3.5 Implementasi Tampilan Manualisasi *Naïve Bayes*

Implementasi tampilan manualisasi *Naïve Bayes* akan menampilkan judul tampilan, perhitungan manualisasi *Naïve Bayes* untuk menampilkan probabilitas diterima calon mahasiswa di tiap fakultas, dan tombol tutup digunakan untuk menutup tampilan manualisasi *Naïve Bayes*. Implementasi tampilan manualisasi pada *Naïve Bayes* dapat dilihat pada Gambar 5.5.

MANUALISASI NAIVE BAYES

Probabilitas Berhasil Teknik = 0.25
 Probabilitas Berhasil Kedokteran = 0.25
 Probabilitas Berhasil Teknik = 0.25
 Probabilitas Berhasil Kedokteran = 0.25
 Data ke 1
 Mean agama terima : 90.19920043945312
 Deviasi Agama terima : 2.4101104718475765
 Likelihood Agm : 0.05868602163238158
 Mean kwn terima : 90.80660003662109
 Deviasi Kwn terima : 2.378146791643631
 Likelihood Kwn : 0.10963568698027511
 Mean Ind terima : 86.52699981689453
 Deviasi Ind terima : 3.67693818020585
 Likelihood Ind : 0.08022759120546408
 Mean ing terima : 89.9933999633789
 Deviasi Ing terima : 2.4651695419489195
 Likelihood Ing : 0.15584955481777557
 Mean sjr terima : 89.87300018310548
 Deviasi Sjr terima : 3.6936006725024715
 Likelihood Sjr : 0.023433926957903865

KEMBALI

Gambar 0.5 Implementasi Tampilan Manualisasi *Naïve Bayes*

5.3.6 Implementasi Tampilan Pengujian *Weighted Product (WP)*

Implementasi tampilan pengujian WP menampilkan judul tampilan, hasil rekomendasi WP menampilkan hasil rekomendasi jurusan untuk calon mahasiswa,

Data	Data Awal	Hasil
Data 1	Data 1	1. industri 2. kimia 3. mesin
Data 2	Data 20	1. industri 2. kimia 3. mesin
Data 3	Data 71	1. mesin 2. industri 3. kimia

Akurasi 46%

MANUALISASI

SELESAI

Gambar 0.6 Implementasi Tampilan Pengujian WP

Implementasi Tampilan Manualisasi *Weighted Product* (V

Implementasi tampilan manualisasi WP menampilkan judul dengan manualisasi WP digunakan untuk menentukan hasil re kuliah berdasarkan hasil klasifikasi fakultas menggunakan me tombol tutup digunakan untuk menutup tampilan manu entasi tampilan manualisasi pada WP dapat dilihat pada Gamb

MANUALISASI WEIGHTED PRODUCT

[3 1 2]
0
[1 3 2]
0
[3 2 1]
=[3, 2, 2, 1, 2, 2, 3, 1, 3, 1, 3]

Implementasi tampilan manualisasi WP menampilkan judul tampilan, perhitungan manualisasi WP digunakan untuk menentukan hasil rekomendasi jurusan kuliah berdasarkan hasil klasifikasi fakultas menggunakan metode *Naïve Bayes*, tombol tutup digunakan untuk menutup tampilan manualisasi WP. Implementasi tampilan manualisasi pada WP dapat dilihat pada Gambar 5.7.

[illegible]

Gambar 0.7 Tampilan Manualisasi WP

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas mengenai pengujian serta analisis terhadap implementasi klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme *Naïve Bayes*- WP.

6.1 Pengujian dan Analisis Jumlah Data Latih dan Data Uji Algoritme *Naïve Bayes*

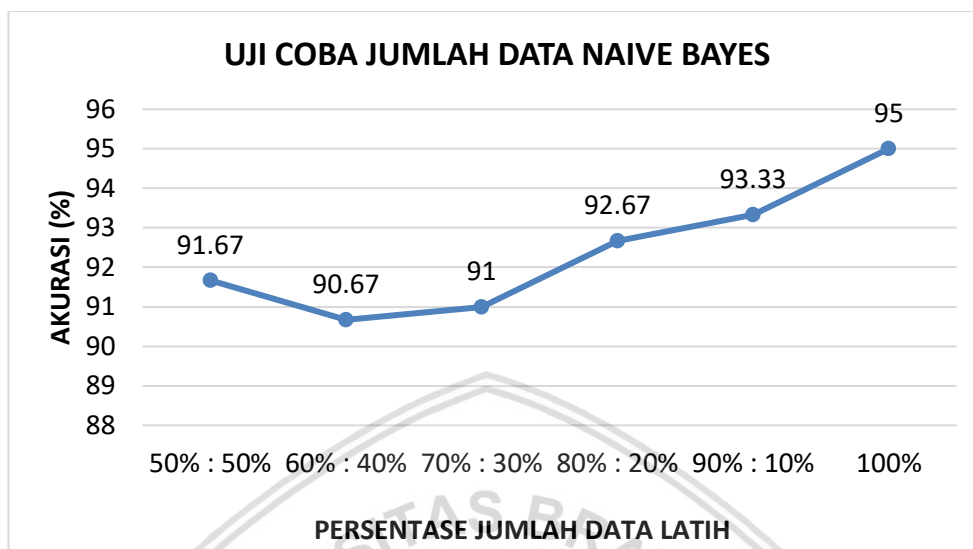
Pada uji coba jumlah data latih dan data uji untuk algoritme *Naïve Bayes* digunakan data pelajar SMAN 3 Malang. Untuk tiap percobaan, digunakan data latih dengan presentase yang berbeda, tujuan untuk mengetahui jumlah pasangan persentase dari data latih dan data uji dengan nilai keakuratan tertinggi. Data uji berasal dari data latih yang tidak digunakan sebagai data latih didalam uji coba. Total data keseluruhan 182 data, dengan batasan pengujian adalah pengujian yang dilakukan berdasarkan skenario yang dilakukan sebanyak 3 kali. Untuk hasil uji coba data latih dan data uji algoritme *Naïve Bayes* dapat dilihat pada Tabel 6.1.

Tabel 0.1 Hasil Uji Coba Jumlah Data Latih dan Data Uji Algoritme *Naïve Bayes*

Jumlah Data Latih (persen)	Jumlah Data Uji (persen)	Percobaan			Rata-Rata Akurasi
		1	2	3	
50	50	92%	94%	89%	91,67%
60	40	93%	91%	88%	90,67%
70	30	89%	96%	88%	91%
80	20	89%	92%	97%	92,67%
90	10	95%	95%	90%	93,33%
100	100	95%	95%	95%	95%

Pada tabel di atas dapat dilihat bahwa rata-rata akurasi terbaik didapat ketika menggunakan seluruh data yang menghasilkan nilai akurasi 95% dan untuk akurasi jumlah pasangan data latih dan data uji yang terbaik menggunakan jumlah data latih sebanyak 90% dari total data dan data uji sebanyak 10% dari total data yang menghasilkan nilai akurasi 93,33%. Jika dilihat pada hasil rata-rata akurasi, dapat dikatakan semakin banyak data latih yang digunakan, semakin tinggi juga rata-rata akurasi yang bisa didapat dari sistem hal ini dikarenakan pada proses pelatihan sistem mengalami pembelajaran yang lebih banyak seiring dengan banyaknya data

latih yang bervariasi sehingga data baru yang diujikan dapat dikelompokkan dengan tepat. Untuk penjelasan rata-rata akurasi secara grafik dapat dilihat pada Gambar 6.1.



Gambar 0.1 Grafik Hasil Rata-Rata Akurasi Jumlah Data Latih dan Data Uji

6.2 Pengujian dan Analisa Jumlah Data Latih dan Data Uji Algoritme *Naïve Bayes* dengan Data Diskrit

Pada uji coba jumlah data latih dan data uji pada algoritme *Naïve Bayes* dengan menggunakan data diskrit, data nilai akademik yang bertipe data kontinu dirubah ke dalam data diskrit untuk mempermudah perhitungan. Pengujian ini bertujuan untuk mengetahui perbandingan akurasi antara sistem yang menggunakan data bertipe kontinu dengan sistem yang menggunakan data bertipe diskrit. Untuk mengubah data kontinu ke dalam bentuk diskrit, digunakan *range* yang dijelaskan pada Tabel 6.2.

Tabel 0.2 Keterangan *Range* Nilai

Range	Nilai Diskrit
Nilai < 70	Rendah
70 <= Nilai <= 85	Sedang
Nilai > 85	Tinggi

Pada data latih contoh perhitungan ini menggunakan data sample yang terdapat pada Tabel 4.4 dan untuk data latih yang terdapat pada Tabel 4.3. . Berdasarkan Tabel 6.2, data diskrit memiliki 3 kategori, yaitu TINGGI (untuk nilai > 85), SEDANG (untuk 70 <= nilai <= 85), dan RENDAH (untuk nilai < 70). Dalam

contoh pengubahan, digunakan data di dalam Tabel 4.4. Nilai mata pelajaran agama pada data pertama memiliki nilai 86, sehingga memiliki nilai diskrit TINGGI karena memiliki nilai lebih dari 85. Untuk hasil lengkap dari pengolahan data menjadi data diskrit dapat dilihat pada Tabel 6.3.

Tabel 0.3 Hasil Pengolaha Data Kontinu ke Data Diskrit

Agama	Kwn	Indonesi	Inggris	Sejarah	Seni	Olahraga	Fisika	Matematika	Kimia	Biologi
Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi
Tinggi	Tinggi	Sedang	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi
Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi
Tinggi	Tinggi	Sedang	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi
Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi
Sedang	Sedang	Sedang	Tinggi	Sedang	Sedang	Tinggi	Sedang	Sedang	Sedang	Tinggi
Tinggi	Tinggi	Sedang	Tinggi	Tinggi	Sedang	Tinggi	Rendah	Tinggi	Tinggi	Tinggi
Tinggi	Tinggi	Sedang	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Sedang	Tinggi	Tinggi
Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Sedang	Sedang	Sedang	Sedang	Tinggi	Sedang
Sedang	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi	Sedang	Sedang	Sedang	Sedang

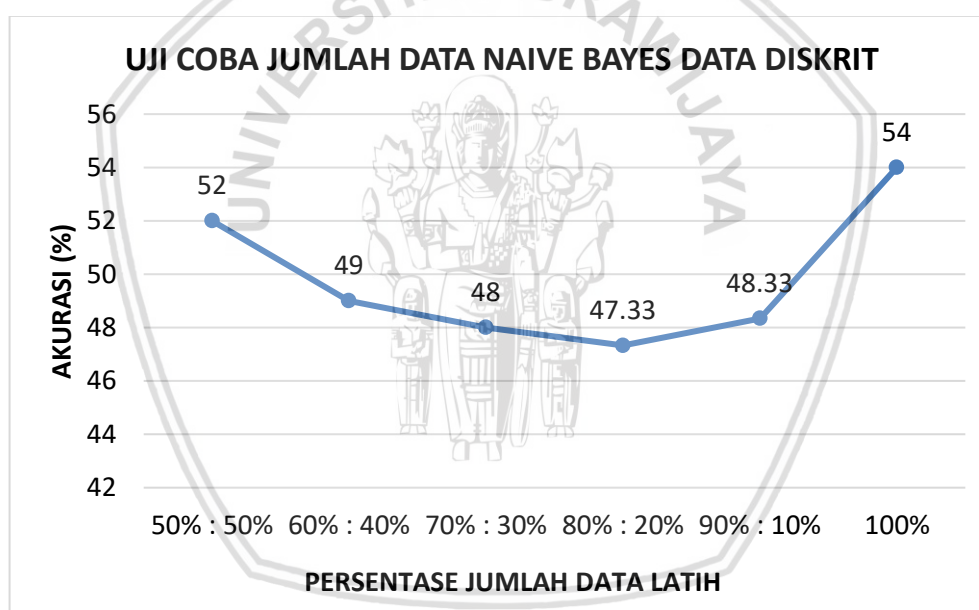
Kemudian setelah dilakukan pengubahan data latih, dilakukan perhitungan untuk menentukan klasifikasi dari data uji untuk perhitungan likelihood, untuk data diskrit digunakan Persamaan 2.5. Setelah mendefinisikan cara perhitungan kemungkinan dengan data latih diskrit, dilakukan uji coba jumlah data latih algoritme *Naïve Bayes* dengan batasan pengujian adalah yang dilakukan berdasarkan skenario yang dilakukan sebanyak 3 kali. Untuk hasil uji coba data latih dan data uji algoritme *Naïve Bayes* dapat dilihat pada Tabel 6.4.

Tabel 0.4 Hasil Uji Coba Jumlah Data Latih dan Data Uji Algoritme *Naïve Bayes* dengan Data Diskrit

Jumlah Data Latih (persen)	Jumlah Data Uji (persen)	Percobaan			Rata-Rata Akurasi
		1	2	3	
50	50	53%	52%	51%	52%
60	40	47%	51%	49%	49%
70	30	44%	53%	47%	48%
80	20	45%	47%	50%	47,33%

90	10	50%	45%	50%	48,33%
100	100	54%	54%	54%	54%

Dari hasil pengujian tersebut, dapat dikatakan bahwa sistem dengan data latih bertipe diskrit memiliki akurasi maksimum 54% yang didapat pada keadaan data latih dan data uji menggunakan seluruh data, untuk pasangan jumlah data latih dan data uji terbaik menggunakan jumlah data latih sebanyak 90% dan total data dan data uji sebanyak 10% dari total data dengan hasil akurasi 48,33%. Rata-rata akurasi ini masih berada di bawah akurasi dari sistem dengan data latih bertipe kontinu. Hal ini dapat disebabkan karena *range* untuk menentukan nilai data diskrit belum tentu *range* dengan nilai yang optimum, sehingga menimbulkan nilai akurasi yang berada di bawah perhitungan dengan tipe data kontinu. Dengan hasil ini, dapat disimpulkan untuk klasifikasi fakultas dengan data nilai, sistem dengan data latih bertipe kontinu lebih bisa diandalkan dibandingkan sistem dengan data latih bertipe diskrit. Untuk penjelasan akurasi secara grafik dapat dilihat pada Gambar 6.2.



Gambar 0.2 Grafik Hasil Rata-Rata Akurasi Uji Coba Jumlah Data dengan Data Latih dan Data Uji Bertipe Diskrit

6.3 Pengujian dan Analisis Pengujian Bobot Nilai Minimum Tiap Jurusan Pada Algoritme *Weighted Product* (WP)

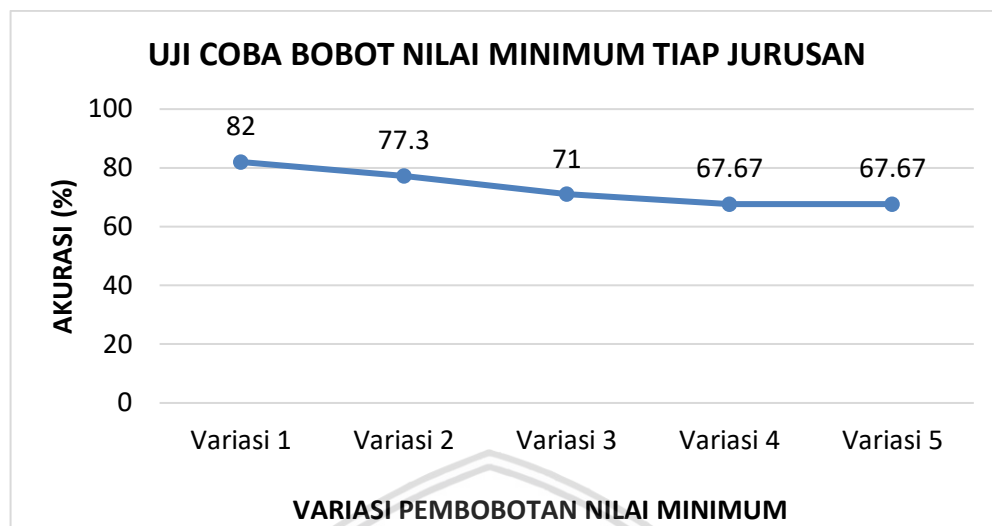
Pada uji coba bobot nilai minimum tiap jurusan pada algoritme WP, dilakukan 3 kali percobaan menggunakan 20 data latih yang berbeda dan dalam perhitungan menggunakan 5 variasi bobotan nilai minimum tiap jurusan. Untuk penjelasan

variasi pembobotan nilai minimum terdapat pada Tabel 4.1. Untuk hasil uji coba pengujian bobot nilai minimum tiap jurusan pada algoritme WP dapat dilihat pada Tabel 6.5.

Tabel 0.5 Hasil Uji Coba Pengujian bobot nilai minimum tiap jurusan pada algoritme WP

Variasi Bobot Nilai Minimum	Percobaan			Rata-Rata Akurasi
	1	2	3	
Variasi 1	90%	65%	91%	82%
Variasi 2	85%	65%	82%	77,3%
Variasi 3	80%	60%	73%	71%
Variasi 4	75%	55%	73%	67,67%
Variasi 5	75%	55%	73%	67,67%

Dari Tabel 6.5 dapat dilihat bahwa dari 3 kali percobaan pada 5 variasi bobot nilai minimum tiap jurusan didapat hasil pasangan bobot yang baik pada variasi ke 1 dengan menggunakan nilai bobot dengan dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 2 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 3 yang paling mendekati dengan nilai calon mahasiswa menghasilkan akurasi 82% dan rata rata akurasi terendah terdapat pada variasi 4 dan 5. Untuk variasi 4 menggunakan nilai bobot dengan nilai dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 5 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 9 yang paling mendekati dengan nilai calon mahasiswa dan variasi 5 dengan menggunakan nilai bobot dengan nilai dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 6 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 11 yang paling mendekati dengan nilai calon mahasiswa yang keduanya menghasilkan akurasi 67,67%. Untuk penjelasan rata-rata akurasi secara grafik dapat dilihat pada Gambar 6.3.



Gambar 0.3 Grafik Hasil Rata-Rata Akurasi Uji Coba bobot nilai minimum tiap jurusan pada algoritme WP

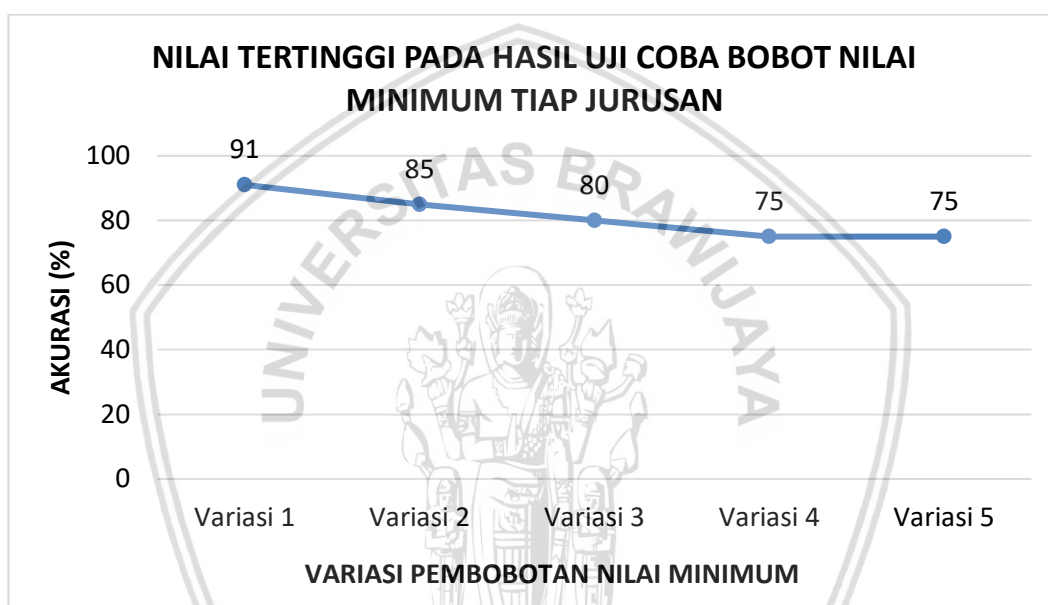
Pada uji coba bobot nilai minimum tiap jurusan pada algoritme WP, dilakukan 3 kali percobaan menggunakan 20 data latih yang berbeda dan dalam perhitungan menggunakan 5 variasi bobot nilai minimum tiap jurusan yang mendapatkan nilai tertinggi pada masing masing variasi dapat dilihat pada Tabel 4.2. Untuk hasil uji coba pengujian bobot nilai minimum tiap jurusan pada algoritme WP yang mendapatkan nilai tertinggi pada masing masing variasi dapat dilihat pada Tabel 6.6.

Tabel 0.6 Hasil Nilai Tertinggi Pada Pengujian bobot nilai minimum tiap jurusan pada algoritme WP

Variasi Nilai Bobot Minimum	Nilai Tertinggi
Variasi 1	91%
Variasi 2	85%
Variasi 3	80%
Variasi 4	75%
Variasi 5	75%

Dari Tabel 6.6 dapat dilihat pada 5 variasi bobot nilai minimum tiap jurusan didapat hasil nilai tertinggi pada tiap variasi dan nilai akurasi tertinggi terdapat pada variasi ke 1 dengan menggunakan nilai bobot dengan nilai dengan

nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 2 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 3 yang paling mendekati dengan nilai calon mahasiswa menghasilkan akurasi 91% dan nilai akurasi terendah terdapat pada variasi 4 dan 5. Untuk variasi 4 menggunakan nilai bobot dengan nilai dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 5 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 9 yang paling mendekati dengan nilai calon mahasiswa dan variasi 5 dengan menggunakan nilai bobot dengan nilai dengan nilai 1 yang paling jauh dengan nilai calon mahasiswa, nilai bobot 6 yang jauh kedua dengan nilai calon mahasiswa, nilai bobot 11 yang paling mendekati dengan nilai calon mahasiswa yang keduanya menghasilkan akurasi 75%. Untuk penjelasan rata-rata tertinggi akurasi esecara grafik dapat dilihat pada Gambar 6.4.



Gambar 0.4 Grafik Hasil Nilai Akurasi Tertinggi Tiap Variasi Pada Uji Coba bobot nilai minimum tiap jurusan pada algoritme WP

BAB 7 PENUTUP

7.1 Kesimpulan

Bedasarkan hasil penelitian mengenai implementasi pada klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA menggunakan algoritme *Naïve Bayes*-WP, didapatkan beberapa kesimpulan, yaitu:

1. Pada algoritme *Naïve Bayes* bersama dengan algoritme WP dapat diimplementasikan untuk klasifikasi dan rekomendasi jurusan kuliah bagi pelajar SMA. Hasil dari penelitian ini adalah penjurusan yang direkomendasikan berdasarkan nilai dari calon mahasiswa. Pada penelitian ini data diklasifikasikan terlebih dahulu ke dalam sebuah fakultas dengan menggunakan metode *Naïve Bayes*, kemudian data yang memiliki kelas diterima disuatu fakultas tertentu akan diproses oleh algoritme WP untuk pemberian rekomendasi jurusan. Pada proses pemberian rekomendasi jurusan melibatkan proses memberi bobot nilai minimum pada tiap jurusan. Penelitian ini menggunakan 5 model variasi bobot nilai minimum pada tiap jurusan. Dari hasil pengujian bobot nilai minimum tiap jurusan pada algoritme WP, didapatkan kesimpulan bahwa hasil rekomendasi jurusan yang terbaik ketika menggunakan variasi 1 dengan nilai pembobotan 1,2 dan 3.
2. Pada implementasi penelitian ini dimulai dari pemilihan fakultas dengan menggunakan algoritme *Naïve Bayes*, dengan rata-rata akurasi terbaik didapat ketika menggunakan seluruh data yang menghasilkan nilai akurasi 95% dan untuk akurasi jumlah pasangan data latih dan data uji yang terbaik menggunakan jumlah data latih sebanyak 90% dari total data dan data uji sebanyak 10% dari total data yang menghasilkan nilai akurasi 93,33%.

7.2 Saran

Bedasarkan kesimpulan yang telah didapat, ada beberapa saran untuk pembaca jika ingin melakukan penelitian lebih lanjut, sebagai berikut:

1. Pada penelitian ini hanya menggunakan 1 SMA atau 1 tempat saja dan menggunakan data 2 tahun terakhir dikarenakan parameter rekomendasi yang diperhitungkan hanyalah berdasarkan nilai. Untuk penelitian selanjutnya disarankan menggunakan data per tahun saja dikarenakan data tiap tahun selalu berbeda dan menambah parameter seperti wawancara kepada calon mahasiswa tersebut. Dan jika memungkinkan, data yang dikumpulkan adalah data yang memiliki universitas tujuan yang sama, sehingga hasil yang didapat lebih baik lagi.

2. Dalam penelitian ini tidak memperhitungkan parameter psikologi dari calon mahasiswa, sehingga dalam penelitian berikutnya dapat ditambahkan perhitungan dengan menambahkan parameter psikologi anak untuk mendapatkan hasil rekomendasi yang lebih baik lagi.



DAFTAR PUSTAKA

- Fartindyyah , N. & S., 2014. Sistem Pendukung Keputusan Peminatan Sma Menggunakan Metode Weighted Product (WP). *Jurnal Kependidikan*, 44(2), pp. 139-145.
- Alfinda, L. A., Soebroto, A. A. & Mardi Putri, R. R., 2014. Sistem Pendukung Keputusan Pemilihan Penerima Jamkesmas Menggunakan Metode Weighted Product. *Doro Jurnal*, Volume 3.
- Alfinda, L. A., Soebroto, A. A. & Putri, R. R. M., 2014. Sistem Pendukung Keputusan Pemilihan Penerima Jamkesmas Menggunakan Metode Weighted Product. *Doro Jurnal*, Volume 3.
- Anon., 2017. *Kementerian Riset Teknologi Dan Pendidikan Tinggi*. [Online] Available at: <https://ristekdikti.go.id/101-906-siswa-lulus-snmptn-2017/> [Diakses 2018 Maret 30].
- B., t.thn. *Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi*, s.l.: Jurnal Penelitian Teknik Informatika .
- Deorah, et al., 2010. SAES-Expert System for Advising Academic Major. *Computer Science and Engineering Department Thapar University*.
- Hill, P. L. et al., 2016. Majoring in Selection, and Minor in Socialization: The Role of the College Experience in Goal Change Post-High School. *Journal of Personality*, 84(2), p. 10.
- Kumala, A. T., Benarkah, N. & Tjandra, E., 2015. Pembuatan Sistem Pendukung Keputusan Pemilihan Jurusan Kuliah Bagi Siswa Sma Berbasis Web Dengan Metode Promethee. *Jurnal Ilmiah Mahasiswa Universitas Surabaya*, Volume 4.
- Lestari, D. I., 2016. Klasifikasi online dan google Irma Devi Lestari. *Iqra'*, 10(2).
- M. et al., 2016. Implementasi Metode Naïve Bayes Classifier Untuk Seleksi Asisten Praktikum Pada Simulasi Hadoop Multinode Cluster. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, Volume 3.
- Pangestu, M. E., 2016. *Sistem Pendukung Keputusan Penerimaan Bpjs Menggunakan Metode Naïve Bayes Di Kabupaten Nganjuk*, Kediri: Fakultas Teknik – Teknik Informatika .
- Prasetya, D. E., Cholissodin, I. & Widodo, A. W., 2016. Sistem Pengklasifikasi Jurusan Kuliah Untuk Pelajar Sma Menggunakan Algoritma Ahp-Naïve Bayes. *Doro Jurnal*, Volume 8.
- Prasetyo, E., 2012. *Data Mining Konsep Dan Aplikasi Menggunakan Matlab*, Yogyakarta: Andi.

Ramdhani, J., 2017. *Masuk PTN Makin Berebut, Jumlah Peserta SBMPTN 2017 Bertambah*. [Online] Available at: <https://news.detik.com/berita/d-3527817/masuk-ptn-makin-berebut-jumlah-peserta-sbmptn-2017-bertambah> [Diakses 30 Maret 2018].

Rudiarsih, N., 2012. *Sistem Pendukung Keputusan untuk Proses Penentuan Rumah Tangga Miskin Menggunakan Metode Weighted Product*. [Online] Available at: <http://filkom.ub.ac.id/doro/archives/detail/DR00026201306> [Diakses 6 November 2017].

Saleh, A., 2015. Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. *Citec*, 2(3).

Salma, T. D., 2016. Sistem Rekomendasi Pemilihan Sekolah Menengah Tingkat Atas Menggunakan Metode Naive Bayes. *Publikasi Ilmiah*.

Sianturi, I. S., 2013. Sistem Pendukung Keputusan Untuk Menentukan Pemilihan Jurusan Siswa Dengan Menggunakan Metode Weighted Product (Studi Kasus: Sma Swasta Hkbp Doloksanggul). *Informasi dan Teknologi Ilmiah (INTI)*, Oktober.1(1).

Wijaya, R., 2014. Perancangan Sistem Pendukung Keputusan Untuk Pemilihan Lokasi Perumahan Menggunakan Weighted Product Method (WPM). *Jurnal Informatika*, 10(1), pp. 61-78.

